

Perpustakaan SKTM

A Graphical Package for Geometry

Name: Hee Lee Li

Matrix No.: WEK000073

Supervisor: Dr. Lee Sai Peck

Moderator: Pn. Raja Jamilah Raja Yusof

ABSTRAK

Laporan Latihan Iimiah bertajuk “ Pakej Bergrafik bagi Geometri”. Latihan ini merupakan sebahagian daripada keperluan Sarjana Muda Sains Komputer, Universiti Malaya. Ia telah dibangunkan menggunakan perisian Java. Aspek-aspek multimedia seperti grafik, animasi dan teks dengan menggunakan perisian Java telah dimasukkan ke dalam pakej sebaik mungkin. Di samping itu, pakej itu juga direkabentuk supaya ianya benar-benar berinteraktif, iaitu pengguna turut melibatkan diri semasa menggunakan pakej. Ia juga memberi tindak balas bersesuaian mengikut kehendak pengguna.

Kandungan pakej ini telah dibuat mengikut Sukatan Pelajaran Matematik KBSM di negara kita. Pakej ini menyediakan nota-nota ringkas serta latihan bagi memperkukuh minda pelajar-pelajar yang menggunakan pakej dari semasa ke semasa mengikut topik tertentu. Di samping itu, sudut latihan juga disediakan untuk membolehkan pelajar mempraktiks sendiri dengan mengubahsuai ciri-ciri geometry asas yang sedia ada atau penghasilan lukisan geometri dari pakej ini.

Metadologi yang digunakan untuk pembangunan perisian ini adalah Pendekatan Unified. Model ini dipilih sebab ia adalah berpandukan use-case (use-case driven) dan tertumpu kepada senibina sistem (Architecture-centric).

Rekabentuk antaramuka sistem adalah berdasarkan prinsip mesra pengguna memandangkan pengguna sasaran adalah pelajar-pelajar sekolah rendah yang berumur 10-12 tahun.

Proses implementasi dimulakan dengan pemasangan (installation) perisian yang diperlukan untuk pembangunan sistem. Ini termasuklah pemasangan JDK 1.3.1 dan JCreator LE 2.0 dan pemastian konfigurasi perkakasan yang sepadan. Selepas itu, Proses pengkodan dimulakan. Kaedah pengkodan dijalankan dengan mengikut algoritma dan beberapa kaedah piawaian yang baik supaya memudahkan proses penyelenggaraan pada masa depan.

Selepas pengimplementasi selesai dijalankan, proses pengujian dijalankan. Pengujian dijalankan secara berperingkat. Jenis-jenis pengujian yang dijalankan termasuklah pengujian unit, pengujian modul, pengujian integrasi, pengujian sistem dan pengujian penerimaan. Perancangan pengujian harus dijalankan untuk menjamin kelancaran proses. Penyelenggaraan juga diperlukan selepas proses pengujian.

Penilaian dijalankan pada pengakhiran pembangunan sistem. Kekuatan dan kekurangan sistem dinilai di mana kebaikan dikekalkan dan kekurangannya harus diperbaiki pada masa depan. Cadangan-cadangan terhadap projek juga dibincangkan. Kemudian kesimpulan daripada pengalaman projek juga diketengahkan.

PENGHARGAAN

Pertama sekali, saya ingin mengucapkan ribuan terima kasih kepada Prof. Madya. Dr. Lee Sai Peck selaku penyelia projek ini telah banyak membantu dan membimbing dari awal hingga akhir projek ini. Segala teguran dan cadangan yang diberikan oleh beliau telah banyak membantu saya dalam menghasilkan “Pakej Bergrafik bagi Geometri”.

Ucapan setinggi-tinggi terima kasih kepada Pn. Raja Jamilah Raja Yusof selaku moderator projek untuk mengawasi projek .

Sekalung budi juga diucapkan kepada pelajar-pelajar dan guru-guru salah sebuah sekolah rendah di Seksyen 17 kerana memberikan kerjasama mengisi boring soal kaji selidik yang disediakan

SENARAI ISI KANDUNGAN

ABSTRAK	i
PENGHARGAAN	iv
SENARAI JADUAL	ix
SENARAI DIAGRAM	xi

BAB 1 PENGENALAN

1.1 Pengenalan	1
1.2 Definasi Projek	2
1.3 Objektif	3
1.4 Masalah Yang Sedia Ada	4
1.5 Kepentingan Projek	5
1.6 Skop dan Kekangan Projek	
1.6.1 Skop Projek	5
1.6.2 Kekangan Projek	6
1.7 Pengguna Sasaran	7
1.8 Penjadualan Projek	7
1.9 Hasil Output Yang Dijangkakan	8
1.10. Ringkasan Setiap Bab	9

BAB 2 KAJIAN LITERASI

2.1 Pengenalan	11
2.2 Apakah itu Geometri?	11
2.3 Kepentingan Geometri	12
2.4 Interaksi Insani Komputer	12
2.5 Pembelajaran Berbantuan Komputer	13
2.6 Arahan Berbantuan Komputer	15
2.7 Kelebihan Penggunaan Komputer Dalam Pendidikan	17
2.8 Perbandingan Sistem Yang Sedia Ada	
2.8.1 Drawing For Children	19
2.8.2 FXDraw	20
2.9 Ringkasan Bab 2	21

BAB 3 METODOLOGI

3.1 Pengenalan	22
3.2 Metodologi Pembangunan	22
3.2.1 Pendekatan Unified	25
3.2.2 Fasa Pendekatan Unified	27
3.3 Fasa Aliran Kerja	28
3.4 UML	30
3.5 Sebab-sebab Pendekatan Unified dipilih.	30
3.6 Ringkasan Bab 3	31

BAB 4 ANALISIS

4.1 Pengenalan	31
4.2 Teknik Mendapat Maklumat	32
4.3 Keperluan Fungsian	33
4.3.1 Rajah kes guna	35
4.4 Keperluan Bukan Fungsian	38
4.5 Pertimbangan Sistem Pengendalian	
4.5.1 Pengenalan	39
4.5.2. Windows 98	40
4.6.3. Windows NT 4.0	41
4.5.4 Windows 2000	41
4.5.5 UNIX	42
4.6 Sistem Pengendalian Yang Dipilih	43
4.7 Pertimbangan Bahasa Pengaturcaraan	
4.7.1 Pengenalan	43
4.7.2 Visual Basic 6	44
4.7.3 Java	45
4.7.4 Matlab	46
4.8 Bahasa Pengaturcaraan Yang Dipilih	48
4.9 Kebaikan Java	49
4.10 Perkakasan Minimum Yang Diperlukan	51
4.11 Perisian Yang Dipilih.	51

4.12 Perisian Integrasi	51
4.13 Ringkasan Bab 4	52
BAB 5 REKABENTUK	
5.1 Pengenalan	53
5.2 Rajah kelas	53
5.3 Rajah berjujukan	62
5.4 Rekabentuk struktur program	70
5.5 Rekabentuk skrin	70
5.6 Rekabentuk sistem	
5.6.1 Struktur sistem	71
5.6.2 Rekabentuk antaramuka sistem	71
5.7 Ringkasan Bab 5	75
BAB 6.IMPLEMENTASI DAN PEMBANGUNAN SISTEM	
6.1 Pengenalan	76
6.2.Persekitaran pembangunan	
6.2.1 Konfigurasi perkakasan	76
6.2.2 Konfigurasi perisian	76
6.3 Pembangunan program	
6.3.1 Membuat tinjauan semula ke atas dokumentasi program	78
6.3.2 Merekabentuk program	78
6.3.3 Pengekoden program	79
6.3.4 Pengujian program	79
6.3.5 Dokumentasi program	79
6.4 Pengekoden	
6.4.1 Pendekatan pengekoden	80
6.4.2.Corak pengekoden	91
6.4.3 Dokumentasi kod	93
6.5 Ringkasan Bab 6	94
BAB 7 PENGUJIAN	
7.1 Pengenalan	95

7.2 Jenis-jenis Ralat	96
7.3. Pengujian unit	97
7.4 Pengujian modul	98
7.5 Pengujian integrasi	99
7.6 Pengujian sistem	100
7.7 Pengujian penerimaan	101
7.8 Perancangan pengujian	101
7.9. Penyelenggaraan	102
7.10 Ringkasan Bab 7	103

BAB 8 PENILAIAN SISTEM DAN CADANGAN

8.1 Pengenalan	104
8.2 Kekuatan sistem	104
8.3 Kelemahan sistem	107
8.4 Peningkatan sistem pada masa depan	107
8.5 Masalah-masalah yang dihadapi	108
8.6 Cadangan-cadangan	109
8.7 Ringkasan Bab 7	110

RUJUKAN

BIBLIOGRARFI

LAMPIRAN

SENARAI JADUAL

Jadual 1.1 Skedul Projek	8
Jadual 4.1 Penerangan tentang nama Kes Guna	37
Jadual 4.2 Penerangan bagi aktor Kes Guna	38
Jadual 5.1 Ciri-ciri asas pada GUI	73
Jadual 5.2 Prinsip-prinsip utama antaramuka pengguna	73
Jadual 6.1 Jenis-jenis perisian yang digunakan dalam pembangunan sistem	77
Jadual 8.1 Ciri-ciri antaramuka yang baik	105
Rajah 5.1 Contoh kelas dalam Cambragean kelas	
Rajah 5.2 Contoh kelas yang menunjukkan hubungan Generalisasi (Generalization)	34
Rajah 5.3 Contoh kelas yang menunjukkan hubungan Pertambahan (Association)	35
Rajah 5.4 Contoh kelas yang menunjukkan hubungan satu kepada banyak	35
Rajah 5.5 Contoh kelas yang menunjukkan hubungan Pengagregatan (Aggregation)	36
Rajah 5.6 Cambragean kelas bagi kelas I	37
Rajah 5.7 Cambragean kelas bagi kelas II	38
Rajah 5.8 Cambragean kelas bagi kelas III	39
Rajah 5.9 Cambragean kelas bagi kelas IV	41
Rajah 5.10 Rajah berajukan bagi kes guna "Browse menu"	42
Rajah 5.11 Rajah berajukan bagi kes guna "Colouring"	43
Rajah 5.12 Rajah berajukan bagi kes guna "Drawing"	44
Rajah 5.13 Rajah berajukan bagi kes guna "Move shape"	45
Rajah 5.14 Rajah berajukan bagi kes guna "Change position of shape"	46
Rajah 5.15 Rajah berajukan bagi kes guna "Calculate perimeter"	47
Rajah 5.16 Rajah berajukan bagi kes guna "Calculate area"	48
Rajah 5.17 Rajah berajukan bagi kes guna "Change shape of shape"	49
Rajah 5.18 Rajah bentuk Antaramuka bagi "Drawing tool" System	50

SENARAI RAJAH

Rajah 2.1 Taksonomi Interaksi Insani Komputer	14
Rajah 2.2 Fasa-fasa Proses Pembangunan CAL	16
Rajah 2.3 Perisian “Drawing For Children”	19
Rajah 2.4 Perisian “FXDraw”	20
Rajah 3.1 Gambarajah menunjukkan Fasa-fasa Pendekatan Unified dan Aliran Kerjanya	25
Rajah 4.1 Contoh Kes Guna	36
Rajah 4.2 Gambarajah Kes Guna bagi Sistem	36
Rajah 5.1 Contoh kelas dalam Gambarajah kelas	53
Rajah 5.2 Contoh kelas yang menunjukkan hubungan Generalisasi (Generalization)	54
Rajah 5.3 Contoh kelas yang menunjukkan hubungan Perhubungan (Association)	55
Rajah 5.4 Contoh kelas yang menunjukkan hubungan Association satu kepada banyak	55
Rajah 5.5 Conton kelas yang menunjukkan hubungan Pengaggregatan (Aggregation)	56
Rajah 5.6 Gambarajah kelas bagi sistem	57
Rajah 5.7 Gambarajah kelas bahagian I	58
Rajah 5.8 Gambarajah kelas bahagian II	59
Rajah 5.9 Gambarajah kelas bahagian III	61
Rajah 5.10 Rajah berjujukan bagi kes guna “Browse notes”	62
Rajah 5.11 Rajah berjujukan bagi kes guna “Colouring”	63
Rajah 5.12 Rajah berjujukan bagi kes guna “Drawing”	64
Rajah 5.13 Rajah berjujukan bagi kes guna “Move shape”	65
Rajah 5.14 Rajah berjujukan bagi kes guna “Change graphics size”	66
Rajah 5.15 Rajah berjujukan bagi kes guna “Calculate perimeter”	67
Rajah 5.16 Rajah berjujukan bagi kes guna “Calculate area”	68
Rajah 5.17 Rajah berjujukan bagi kes guna “Change shape arc”	69
Rajah 5.18 Rekabentuk Antaramuka bagi “DrawingTool” Sistem	70

BAB 1 PENGENALAN

1.1 Pengenalan

Setelah memasuki alaf baru ini, apabila kita sibuk memperkatakan tentang Sekolah Bestari yang merupakan salah satu daripada tujuh aplikasi perdana Koridor Raya Multimedia (MSC), maka ia akan berkaitan rapat dengan sistem pembelajaran berkomputer. Dan apabila kita membincangkan tentang sistem pembelajaran berkomputer pula, ia akan dikaitkan pula dengan alat-alat bantuan belajar yang serba moden dan berteknologi. Inovasi teknologi dalam pengurusan bilik darjah membolehkan pelbagai pendekatan dan strategi pembaharuan dan penambahan berlaku. Inovasi teknologi dalam pelajaran guru seperti menggunakan pelbagai jenis teknologi pendidikan dan teknologi matematik boleh menjadikan pengajaran dan pembelajaran lebih inovatif dan kreatif. Penggunaan pelbagai alat gambar boleh menjadikan pengajaran lebih menarik dan kreatif. Pengajaran berpanduan komputer di dalam bilik darjah boleh meluaskan daya kreatif pelajar.

Di samping itu, kemahiran menggunakan komputer dari bilik darjah akan membolehkan pelajar mendapatkan maklumat dan meluaskan domain kreativiti mereka. Teknologi boleh dikira sebagai alat berstrategi dan pendekatan yang mudah untuk meningkatkan daya kreatif pelajar dan guru di bilik darjah. Teknologi yang dapat meluaskan unsur-unsur dan domain kreativiti sama ada dari segi pengajaran mahupun pembelajaran.

Kreativiti dapat dikembangkan dengan lebih dinamik dalam budaya belajar dan budaya mengajar melalui penggunaan informasi matematik dalam geometri dan multimedia. Alat-alat bantuan belajar dan mengajar berteknologi terdiri daripada alatan pandang dengan cakera padat, komputer peribadi dan pelbagai alat bantuan belajar lain yang berunsurkan multimedia.

Dengan adanya alat-alat ini, pelajar akan berminat dan menumpukan perhatian yang lebih semasa pembelajaran diadakan. Pembelajaran yang berasaskan kepada teknologi multimedia ini akan menghasilkan generasi muda yang pintar, inovatif dan kreatif yang boleh menerajui misi negara dalam pencapaian Wawasan 2020.

1.2 Definasi Projek

“Pakej Bergrafik bagi Geometri” ini merupakan satu perisian “stand-alone” yang dibangunkan dengan menggunakan perisian Java tanpa penggunaan pangkalan data dan khas untuk membantu kanak-kanak sekolah yang berumur 10-12 tahun dalam mengenali beberapa geometri asas yang penting dalam kehidupan seharian. Dalam pakej ini, dibekalkan satu modul yang menerangkan beberapa bentuk geometri asas dalam dua dimensi dan tiga dimensi seperti bulatan, segitiga, segiempat, sfera, kuboid, kubus dan sebagainya serta beberapa ciri-ciri asas bentuk-bentuk geometri ini seperti formula mengira luas, perimeter dan isipadu.

Pakej bergrafik ini juga terdapat modul latihan yang menyediakan contoh-contoh asas kepada kanak-kanak dalam mengenali bentuk-bentuk asas geometri, pengiraan perimeter dan luas bagi bentuk-bentuk tertentu.

Di samping itu, pakej ini juga membekalkan satu “peralatan melukis” (drawing tool) yang membolehkan kanak-kanak menghasilkan beberapa bentuk geometri asas berdasarkan kreativiti mereka masing-masing. Kanak-kanak boleh menggunakan tetikus untuk menghasilkan lukisan, menwarnakan lukisan mereka serta mengubahsuai ciri-ciri geometri tersebut. Luas dan perimeter bentuk-bentuk tersebut juga dapat dikira dengan menginputkan nilai-nilai mengikut kesesuaian.

Terdapat modul keluar disediakan dalam setiap antaramuka jika mereka ingin keluar dari aplikasi ini.

Pakej perisian ini dibangunkan dengan menggunakan pakej “Abstract Windowing Toolkit”

(AWT), pakej Swing dan beberapa pakej subordinat yang terdapat dalam Java.

1.3 Objektif

Berikut merupakan beberapa objektif bagi pembangunan perisian “Pakej Bergrafik bagi Geometri”:

1. Membangunkan satu pakej pembelajaran yang mempunyai contoh-contoh asas dan membekalkan kebolehan melukis geometri asas dalam 2 dimensi dan 3 dimensi bagi kanak-kanak sekolah berumur 10-12.
2. Selaras dengan hasrat kerajaan dalam konsep “Sekolah Bestari” iaitu menggalakkan pelajar-pelajar menggunakan komputer dalam proses pembelajaran mereka.
3. Menyediakan dan membiasakan kanak-kanak kepada era pengkomputeran. Kanak-kanak ini akan terdedah kepada banyak aplikasi dalam komputer semasa dewasa kelak.
4. Menyediakan antaramuka yang mesra pengguna dan dapat menarik minat kanak-kanak untuk belajar.
5. Menyediakan capaian yang cepat di mana pengguna boleh mendapat maklumat menggunakan teknik berkomputer tanpa perlu merujuk kepada buku-buku teks atau buku panduan.

6. Mengetengahkan satu perisian pendidikan yang menekankan geometri sebagai teras utama kerana perisian pendidikan matematik bagi sekolah rendah yang ada di pasaran adalah terhad.
7. Memastikan kemantapan generasi muda dalam geometri yang merupakan sebahagian daripada subtopik matematik demi menghadapi cabaran-cabaran masa depan dalam bidang sains dan teknologi.
8. Mewujudkan pemikiran logik di kalangan pelajar dalam menyelesaikan sesuatu masalah.
9. Menggunakan konsep dan formula geometri dalam matematik supaya pelajar dapat mempraktikkannya untuk menyelesaikan masalah harian manusia sejagat pada hari kelak.

1.4 Masalah Yang Sedia Ada

Berdasarkan kajian yang telah dijalankan didapati kebanyakan pelajar pada masa kini menghadapi masalah dalam mempelajari matematik (geometri merupakan sebahagian daripada pelajaran Matematik). Beberapa masalah yang dihadapi oleh pelajar telah dikenalpasti bagi mencari cara alternatif untuk meningkatkan tahap penguasaan pelajar terhadap matapelajaran matematik.

Antara masalah-masalah yang dihadapi ialah:

- Penggunaan buku teks membosankan sesetengah pelajar dan mereka kehilangan minat untuk mempelajari matematik.
- Sikap segelintir pelajar yang menganggap matematik sebagai suatu matapelajaran yang sukar untuk dipelajari menyebabkan mereka tidak

mengambil berat untuk memperbaiki penguasaan mereka dalam matapelajaran tersebut.

- Kaedah pengajaran yang digunakan oleh guru gagal untuk menarik minat pelajar dan ia membebankan pelajar-pelajar untuk mengingat konsep dan kemahiran penting yang diajarkan
- Pelajar gagal untuk memahami konsep sesuatu topik matematik yang diajar dalam penggunaan harian kerana tiada pendedahan secara meluas tentang konsepnya.

1.5 Kepentingan Projek

Pelajar akan diperkenalkan dengan teknik pembelajaran yang lebih menarik dan berkesan memandangkan pakej ini adalah berkonsepkan kreativiti. Pembekalan latihan dan satu set “peralatan” melukis dengan menentukan ciri-ciri asas sesuatu geometri dalam perisian ini berupaya memupuk kemahiran secara kreatif dan kritis bagi menguasai pengetahuan ini.

1.6 Skop dan Kekangan Projek

1.6.1 Skop Projek

Perisian “Pakej Bergrafik bagi Geometri” yang dibangunkan ini akan menfokus kepada kanak-kanak berumur 10-12 tahun. Skop projek ini meliputi kajian tentang jenis antaramuka, metafor dan interaksi yang sesuai dengan psikologi kanak-kanak dalam pembelajaran geometri dengan menggabungkan konsep mental kreativiti dalam sistem pembelajaran mereka. Kajian ini termasuklah pengumpulan maklumat, penganalisaan maklumat, pemahaman maklumat, dan pengujian prototaip.

Skop sistem ini adalah untuk memastikan sistem yang dibangunkan memenuhi keperluan projek. Sistem yang dibangunkan patut:

- Menghasilkan satu modul pembelajaran untuk mengenal beberapa geometri asas serta ciri-cirinya.
- Mempunyai contoh-contoh ringkas mengenai maklumat asas berkaitan dengan geometri asas untuk menyenangkan kanak-kanak mula membelajarinya atau membuat rujukan.
- Menggabungkan cara pembelajaran mental kreativiti iaitu dengan kaedah membekalkan antaramuka kepada kanak-kanak untuk menghasilkan geometri sendiri, mewarnakannya dan mengubahsuai beberapa ciri-ciri dalam bentuk geometri seperti sudut, luas dan perimeternya.
- Mempunyai ciri-ciri yang interaktif dan secara manipulasi terus pada pada aplikasi yang akan dibangunkan tersebut memandangkan kebanyakan penggunaanya terdiri daripada kanak-kanak.

1.6.2 Kekangan Projek

Berikut merupakan kekangan-kekangan dalam perisian yang dihasilkan:

- Pakej perisian ini bukan pakej perisian multimedia keseluruhan kerana tidak ada audio yang dibekalkan.
- Pakej ini hanya dapat mengira luas dan perimeter untuk beberapa bentuk geometri sahaja dan bukan semua geometri yang terdapat dalam nota ringkas disebabkan mengikut kesesuaian silibus dalam matematik para pelajar dalam lingkungan umur 10-12 tahun.

1.7 Pengguna Sasaran

Pengguna sasaran adalah pelajar-pelajar sekolah rendah yang berumur 10-12 tahun atau setara dengan murid-murid Tahun 4, Tahun 5 dan Tahun 6. Ini disebabkan nota dan latihan yang disediakan dalam sistem ini berdasarkan kandungan subjek Matematik UPSR bagi sukatan KBSM. Tambahan pula, peralatan melukis yang dibekalkan dalam sistem itu hanya dapat melukis geometri-geometri asas yang sesuai dengan tahap pelajar-pelajar golongan tersebut.

1.8 Hasil Output Yang Dijangkakan

Hasil output yang dijangkakan adalah:

- Menjimatkan kos
 - Melaksanakan penyelesaian “paperless” dalam proses pengajaran dan pembelajaran, maka akan menjimatkan kos dalam pembelian buku rujukan.
- Mudah dan senang digunakan oleh pengguna
 - Mempunyai antara muka yang menarik dan mempunyai sifat kebolegunaan yang tinggi, dengan itu pengguna senang memahami ciri-ciri antaramuka.
- Menjimatkan masa dalam pembelajaran
 - Dengan penerangan ringkas yang dibekalkan diharapkan dapat para pelajar dapat menjimatkan masa pembelajaran kerana kanak-kanak lebih cepat memahami sesuatu dengan dibekalkan gambar-gambar.
- Memupuk minat yang lebih dalam matapelajaran matematik
 - Dengan berbekalkan gambarajah geometri yang berwarna-warni , diharapkan dapat memupuk minat pelajar yang mendalam dalam pembelajaran matematik.

• 1.9 Penjadualan Projek

Fasa/Masa	06/02	07/02	08/02	09/02	10/02	11/02	12/02	01/03	02/03
Permulaan(Inception-1)									
Permulaan(Inception-2)									
Penghuraian(Elaboration1)									
Penghuraian(Elaboration2)									
Pembinaan(Construction1)									
Pembinaan(Consruction-2)									
Perubahan(Transition-1)									
Perubahan(Transition-2)									

Jadual 1.1 Skedul projek

1.10 Ringkasan Setiap Bab

Bab 1

Bab 1 merupakan bab permulaan yang menerangkan mengenai tajuk projek, objektif, skop dan perancangan pembangunan sistem.

Bab 2

Bab 2 boleh dipecahkan kepada beberapa bahagian yang menerangkan mengenai kajian yang dijalankan, pemerhatian dan tinjauan yang dibuat serta sistem yang sedia ada di pasaran dan internet.

Bab 3

Bab 3 akan diterangkan secara ringkas mengenai metodologi pembangunan pakej pembelajaran bergrafik itu yang menggunakan konsep Pendekatan Unified.

Bab 4

Bab 4 ini akan menerangkan keperluan fungsian dan keperluan bukan fungsi serta fasa kajian awal dan analisis yang dilakukan. Di samping itu, analisis juga dijalankan untuk perbandingan bahasa pengaturcaraan dan sistem pengendalian yang akan digunakan.

Bab 5

Bab 5 ini akan menerangkan tentang rekabentuk antaramuka sistem berdasarkan diagram kelas serta prinsip-prinsip antaramuka yang harus dipatuhi dalam merekabentuk antaramuka yang dihasilkan.

Bab 6

Bab ini menerangkan tentang pengimplementasian yang dijalankan dalam pembangunan sistem. Ini termasuk langkah pemasangan JDK 1.3.1 dan perisian JCreator LE 2.0 serta langkah-langkah pengekodan.

Bab 7

Jenis-jenis pengujian yang dijalankan terhadap sistem akan diterangkan dalam bab ini. Ini termasuklah pengujian unit, pengujian modul, pengujian integrasi, pengujian sistem dan pengujian penerimaan. Setiap pengujian akan diterangkan dalam bab ini dan bagaimana ia dipraktikkan kepada sistem yang dibangun ini.

Bab 8

Bab ini akan menerangkan kekuatan dan kekurangan sistem yang telah dibangun. Selain itu, masalah-masalah yang dihadapi dan cara penyelesaian semasa pembangunan sistem juga dinyatakan. Peningkatan terhadap sistem pada masa depan juga dibincangkan dalam bab ini. Cadangan-cadangan dan kesimpulan terhadap projek juga dibincangkan pada akhir projek.

BAB 2 KAJIAN LITERASI

2.1 Pengenalan

Kajian literasi projek merupakan kajian ilmiah yang dijalankan sebelum sesuatu projek boleh dibangunkan.

Kajian literasi mempunyai beberapa tujuan utama, di antaranya:

- Dapat membantu pembangun sistem mengetahui ciri-ciri sistem yang sedia ada yang mempunyai persamaan dengan sistem yang dibangunkan.
- Untuk mengumpulkan maklumat serta memperolehi pemahaman yang lebih jelas berkenaan sistem yang akan dibangunkan.
- Adalah tidak berguna membangunkan suatu sistem yang telah sedia ada. Pembangun sistem dapat memberi tumpuan kepada sistem yang sedia ada dengan membuat kajian terhadap sistem dan cuba menambah ciri-ciri atau mengubahsuai ciri-ciri yang sedia ada.
- Menyediakan pembangun dengan pengetahuan tentang kekuatan dan kekangan bagi beberapa jenis peralatan pembangunan sistem. Ini dapat membantu pembangun sistem memilih peralatan dan metod yang betul untuk pembangunan sistem.

2.2 Apakah itu Geometri?

Geometri merupakan satu pembelajaran mengenai bentuk yang asalnya digunakan untuk tinjauan dan kajian mengenai tanah, merekabentuk bangunan, dan mengukur isipadu. Apa yang paling penting adalah geometri merupakan asas untuk melihat bentuk perhubungan dan konsep matematik. Tambahan pula, kebanyakan bidang matematik menggabungkan konsep dan pandangan geometri pada asasnya. Contohnya, Descartes, seorang ahli matematik menggunakan

geometri analitik untuk menunjukkan garis dan lengkung yang berkaitan dengan “equations algebraic”. Equation yang berbeza mempunyai formulasi dan ruang fasa geometri yang semulajadi.

Pada masa kini, geometri merangkumi banyak langkah-langkah, teknik dan teori yang berbeza. Ini termasuklah geometri Euclidean dan non-Euclidean, geometri projektif, geometri finite, geometri transformasi, geometri “computational”, geometri “differential”, geometri diskrit, “tilings” dan teori knot.

Geometri Euclidean mendominasi perkembangan geometri di negara barat menggunakan kompas dan “straightedge” untuk mengkaji ketam yang rata dan ruangan tiga dimensi.

Geometri non-Euclidean pula memandangkan ruang bentuk-bentuk asas tidak rata tetapi melengkung. Kebanyakan bentuk di persekitaran kita adalah lengkung dan Jim Casey, seorang ahli matematik berjaya mempraktikkan idea ini untuk menunjukkan bagaimana jurutera dapat menggunakan geometri Riemannian untuk memahami bentuk-bentuk yang melengkung.

Geometri analitikal pula mengintergrasikan algebra dan geometri, ini membenarkan geometri menggunakan sepenuhnya peralatan algebra.

Baru-baru ini, pelbagai kajian sedang dijalankan ke atas geometri dengan menggabungkan pelbagai teori seperti teori knot, “tilings”, teori geometri diskrit, geometri computational dan teori graf untuk menghasilkan pengukuran yang lebih baik pada masa depan untuk menghasilkan aplikasi-aplikasi yang lebih baik. [Catherine A.Gorini, Geometry At Work]

2.3 Kepentingan Geometri

Beberapa kepentingan geometri telah dikaji adalah seperti berikut:

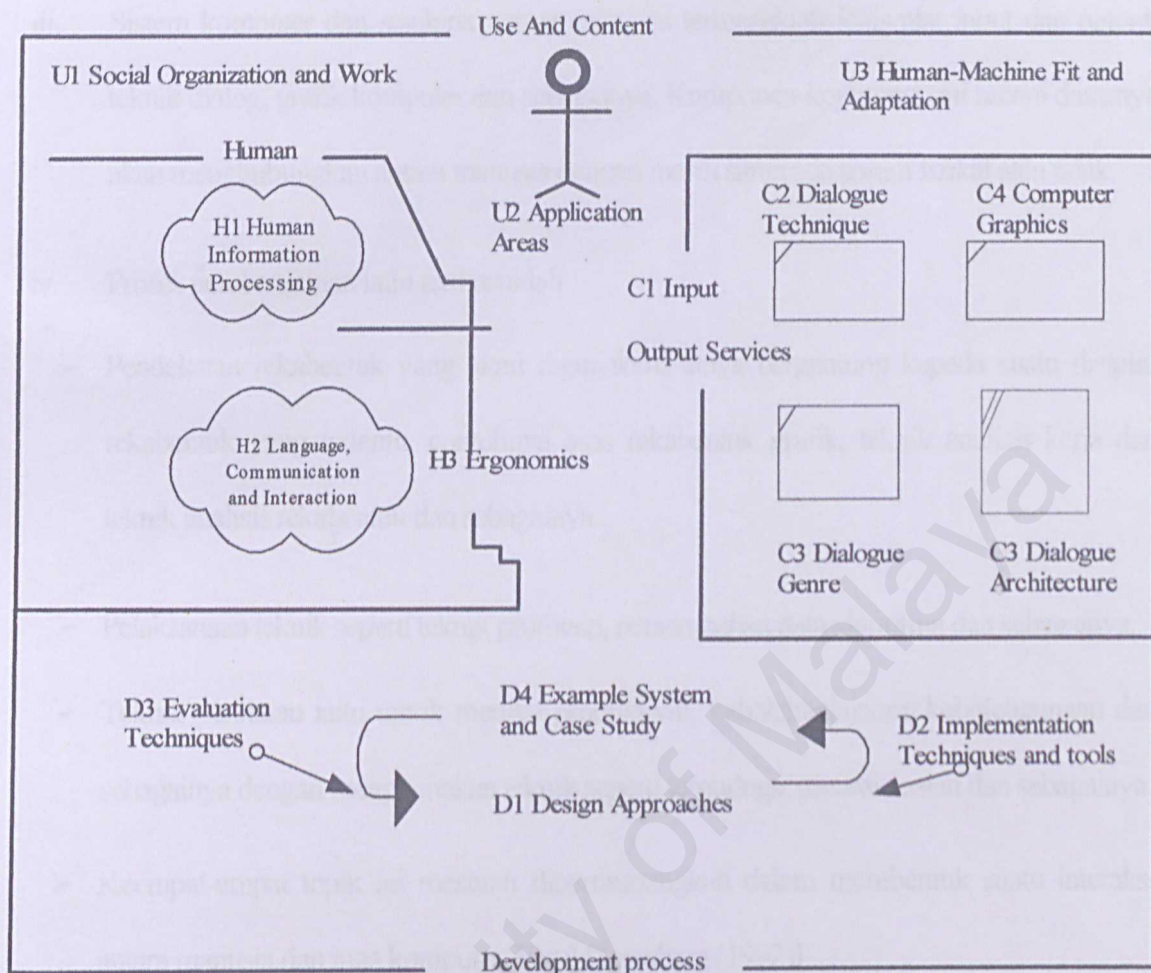
Geometri digunakan untuk menganalisis bentuk yang wujud di persekitaran kerana kebanyakan benda atau bahan yang wujud di persekitaran mempunyai bentuk yang tersendiri.

Geometri adalah penting untuk mensimulasi alam persekitaran. Geometri merupakan peralatan utama bagi jurutera untuk membangun pelan komunikasi dan rekabentuk bangunan, mesin dan sebagainya.

Geometri juga digunakan untuk menghasilkan bentuk dan struktur dari hubungan yang abstrak dan kemudian digunakan untuk menganalisis bentuk tertentu dengan menggunakan teknik geometrical. [Catherine A.Gorini, Geometry At Work]

2.4 Interaksi Insani-Komputer

Interaksi Insani-Komputer (IHK) merupakan satu disiplin yang melibatkan rekabentuk, penilaian dan pelaksanaan bagi interaksi sistem komputer untuk kegunaan manusia dengan mempertimbangkan fenomena utama yang mengelilinginya. Rajah 2.1 merupakan taksonomi bagi IHK yang menerangkan hubungkait antara manusia dengan komputer.



Rajah 2.1 Taksonomi Interaksi Insani-Komputer

Berdasarkan kepada Rajah 2.1, taksonomi IIK boleh dibahagikan kepada empat topik yang utama iaitu:

- Kegunaan dan konteks komputer dalam sesebuah masyarakat. Ini termasuklah kegunaan, tanggapan, kehendak dan pandangan masyarakat terhadap teknologi komputer.
- Ciri-ciri manusia iaitu bagaimana bekerja atau berfikir dan apa yang mereka kehendaki apabila berinteraksi dengan mesin. Ini termasuklah memahami bagaimana manusia memproses maklumat, menstrukturkan tindakan, berinteraksi sesama mereka dan juga mengetahui keperluan fizikal dan psikologikal mereka.

iii. Sistem komputer dan senibina antaramuka, ini termasuklah jenis alat input dan output, teknik dialog, grafik komputer dan sebagainya. Komponen-komponen ini secara dasarnya akan menghubungkan antara manusia dengan mesin sama ada secara fizikal atau tidak.

iv. Proses pembangunan iaitu termasuklah :

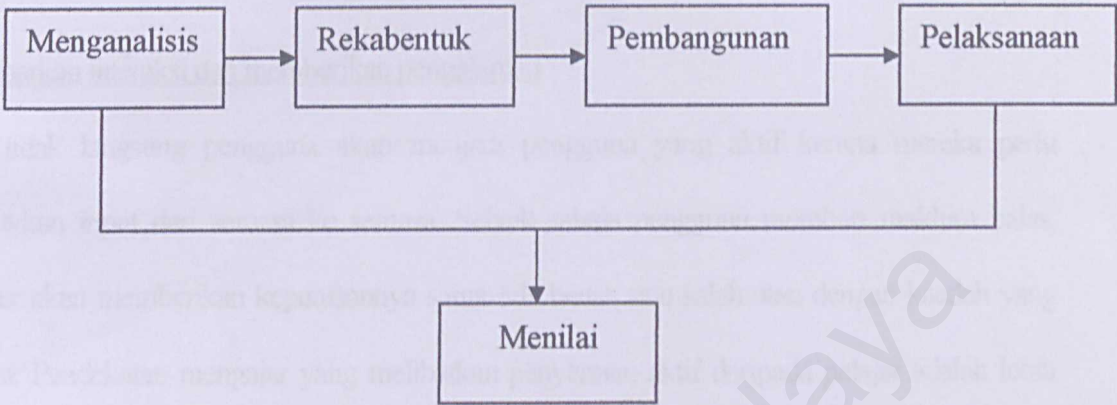
- Pendekatan rekabentuk yang akan digunakan. Ianya bergantung kepada suatu disiplin rekabentuk yang tertentu, contohnya asas rekabentuk grafik, teknik analisis kerja dan teknik analisis rekabentuk dan sebagainya.
- Pelaksanaan teknik seperti teknik prototaip, persembahan data algoritma dan sebagainya.
- Teknik penilaian iaitu untuk menilai produktiviti, kebolehbelaian, kebolegunaan dan sebagainya dengan menggunakan teknik seperti temuduga, senarai soalan dan sebagainya.
- Keempat-empat topik ini mestilah dipertimbangkan dalam membentuk suatu interaksi antara manusia dan juga komputer. [Saul Greenberg (1992)]

2.5 Pembelajaran Berpanduan Komputer (Computer Assisted Learning)

Pembelajaran Berpanduan Komputer (CAL) merujuk kepada penggunaan komputer sebagai sumber pengetahuan untuk membantu para pelajar dalam pembelajaran. CAL menggunakan perisian komputer untuk memudahkan proses pelajaran dan pembelajaran dengan melibatkan teks interaktif, soalan dan tindak balas daripada alat masuk komputer untuk memberi jawapan.

Proses bagi pembangunan CAL boleh dibahagikan kepada lima fasa umum iaitu Analisis, Rekabentuk, Pembangunan, Pelaksanaan dan Menilai. Kadang-kala fasa-fasa ini bertindih dan berkait antara satu sama lain. Namun, ianya memberikan satu garis panduan yang dinamik serta fleksibel bagi membangunkan CAL yang efektif dan efisien.

Gambarajah di bawah menunjukkan hubungan bagi fasa-fasa yang terlibat dalam pembangunan CAL.



Rajah 2.2 Fasa-fasa proses pembangunan CAL

Ciri-ciri CAL:

Pendekatan satu ke satu

Komputer menyediakan interaksi satu ke satu dalam proses mengajar. Ini membolehkan CAL digunakan untuk tujuan pembelajaran tanpa bantuan guru. Pelajar boleh membuat latihan ke atas topik-topik yang dikehendaki pada bila-bila masa sahaja.

Maklum balas cepat atau segera

Bagi CAL adalah dicadangkan supaya menggunakan sekurang-kurangnya mesin 486 bagi persekitaran komputer peribadi. Komputer ini boleh menganalisis masukan yang diberikan dan melaksanakan serta memberi maklum balas secepat mungkin.

Mengawal pencapaian

Selain daripada semakan semula, CAL juga mempunyai ciri-ciri kebolehan pengawasan pencapaian. Misalnya komputer boleh menyimpan rekod-rekod markah pelajar bagi mengawasi

peningkatan tahap penguasaan seorang pelajar terhadap sesuatu topik. Pelajar-pelajar boleh menilai prestasi mereka dalam topik tersebut dan membuat keputusan sama ada ingin melanjutkan ke topik seterusnya atau mengulang semula topik tersebut.

Membenarkan interaksi dan memberikan pengalaman

Secara tidak langsung pengguna akan menjadi pengguna yang aktif kerana mereka perlu memasukkan input dari semasa ke semasa. Sebaik sahaja pengguna memberi maklum balas, komputer akan memberikan keputusannya sama ada benar atau salah atau dengan kaedah yang berlainan. Pendekatan mengajar yang melibatkan penyertaan aktif daripada pelajar adalah lebih berkesan daripada pemerhatian pasif seperti mana yang berlaku di dalam bilik darjah. [Rao.G.S., Rao, A. K, Zoraini Wati Abas, Wan Fauzy Wan Ismail, 'Pembelajaran Berbantuan Komputer', 1991]

2.6 Arahan Berbantuan Komputer (Computer Assisted Instruction)

Era perkembangan teknologi terkini semakin mempengaruhi setiap bidang kehidupan. Aspek teknologi telah digabungkan dalam pendidikan kurikulum sekolah sebagai satu daya usaha ke arah menyemai dan memupuk minat serta sikap positif terhadap perkembangan teknologi. Pendidikan komputer dalam kurikulum melibatkan tiga peranan penting iaitu:

- Pembelajaran tentang perisian, perkakasan dan pengaturcaraan komputer.
- Penggunaan komputer untuk menyelenggara dan memproses data.
- Penggunaan komputer untuk membantu proses pengajaran dan pembelajaran.

CAI sebenarnya adalah akronim bagi Computer-Assisted Instruction. Pada akhir tahun 50-an dan awal tahun 60-an. Penggunaan komputer sebagai peranti arahan hanyalah satu idea yang sedang diperkembangkan oleh segolongan saintis telah melaksanakan idea ini. CAI dapat ditakrifkan

sebagai proses pembelajaran terus yang melibatkan penggunaan komputer dan boleh pengajaran di dalam bentuk mod interaktif bagi menyediakan dan mengawal persekitaran pembelajaran. Mod interaktif ini biasanya dipecahkan kepada “drill” dan latihan, tutorial, simulasi dan permainan, penyelesaian masalah dan pertanyaan.

Kaedah CAI telah digunakan dalam pembelajaran yang meliputi banyak aspek dan bidang tertentu. Antara bidang yang paling banyak melibatkan CAI ialah Sains dan Matematik. Aspek yang paling penting dalam penggunaan CAI di dalam proses pembelajaran ialah ia membolehkan pengguna menjadikan pembelajaran secara interaktif. Para pelajar turut melibatkan diri dalam proses ini dan tidak hanya menjadi pemerhati sahaja. Perisian CAI bukan sahaja boleh dilaksanakan untuk cara berarahan individu tetapi ia juga berkesan jika dibangunkan untuk tujuan pembelajaran secara berkumpulan di mana pelajaran boleh bekerjasama.

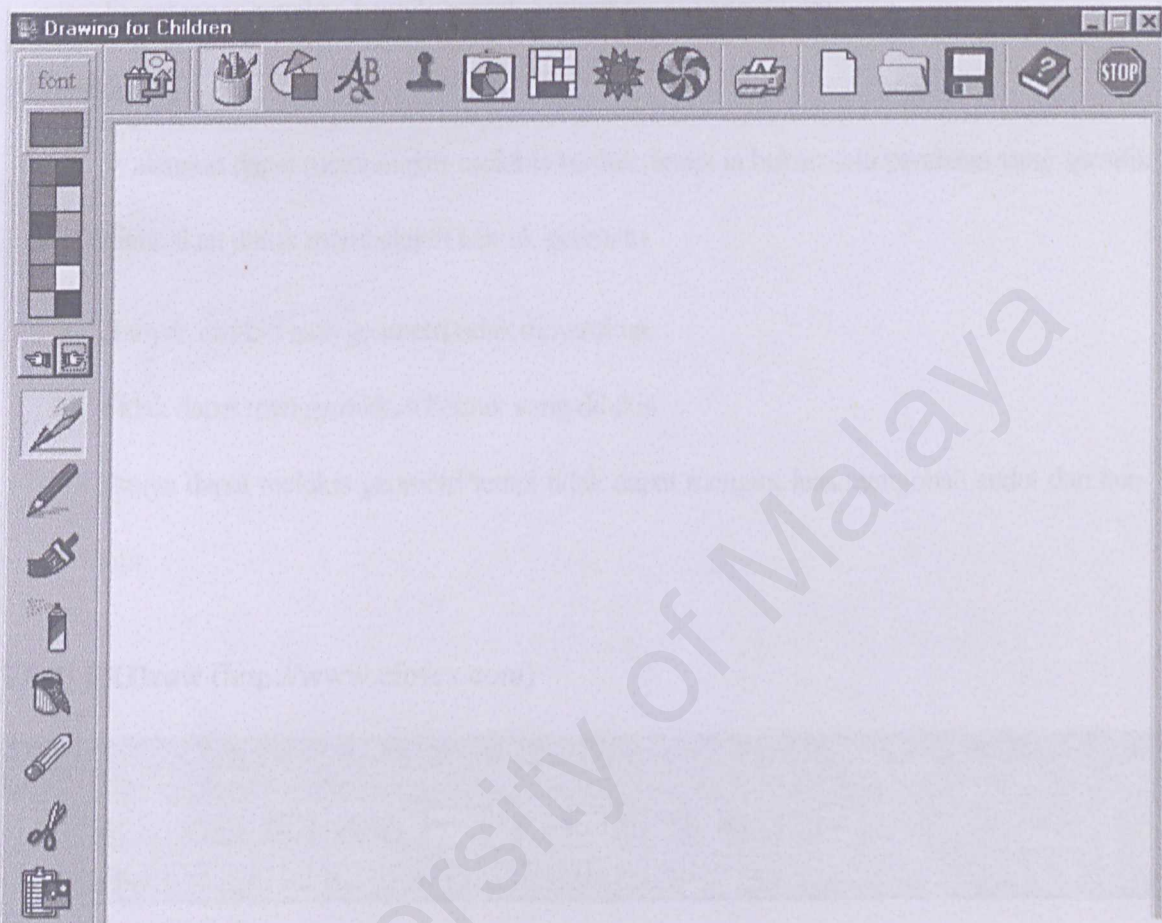
Di samping itu, kejayaan kaedah CAI digunakan dalam proses pembelajaran disebabkan komputer boleh digunakan untuk tujuan penghantaran arahan dalam semua subjek pelajaran sebarang julat umur dan jenis golongan pelajar. [Nievergelt, J., Ventura, A., Hinterberger, H., 1986]

2.7 Kelebihan Penggunaan Komputer dalam pendidikan

Penggunaan komputer dalam matapelajaran Matematik akan dapat memperkembangkan dan memperkayakan lagi tajuk yang diajar. Mesej yang akan disampaikan akan lebih jelas dan menarik. Dengan yang demikian ia akan dapat menarik minat pelajar. Tambahan pula dengan penggunaan multimedia yang dimanipulasikan sepenuhnya akan dapat meningkatkan pemahaman dan seterusnya memperkukuhkan lagi daya ingatan pelajar.

2.8 Perbandingan Sistem Yang Sedia Ada

2.8.1 Drawing For Children (<http://www.cs.uu.nl/~markov/kids/draw/>)



Rajah 2.3 Perisian “Drawing For Children”

Gambarajah di atas merupakan salah satu perisian melukis yang dapat dimuatturun dari internet.

Perisian ini dinamakan “Drawing For Children”.

Ciri-ciri utama sistem:

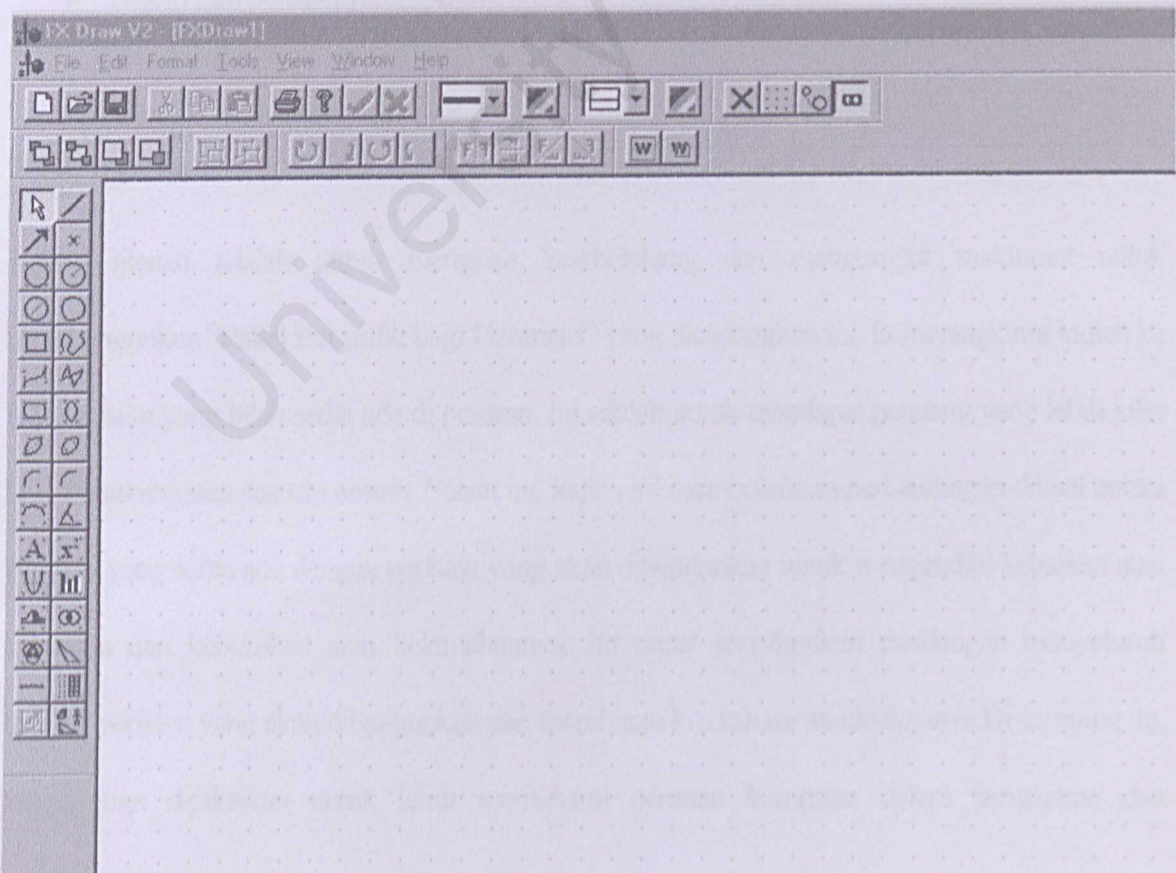
- Sesuai digunakan untuk semua peringkat umur kanak-kanak.
- Di samping dapat melukis bentuk asas geometri, pengguna juga dapat memuatkan semua gambar-gambar kesukaan yang dibekalkan, termasuk bangunan, binatang, serangga, corak dan lain-lain.

- Kanak-kanak dapat melukis dan memilih corak-corak dengan hanya menggunakan tetikus.
- Dapat menwarnakan bentuk-bentuk geometri asas yang dilukis.

Keburukan:

- Walaupun dapat mempelajari melukis bentuk, tetapi ia bukan satu peralatan yang spesifik digunakan untuk membelajari bentuk geometri.
- Banyak ciri-ciri asas geometri tidak dinyatakan.
- Tidak dapat menggerakkan bentuk yang dilukis.
- Hanya dapat melukis geometri tetapi tidak dapat mengira luas, mengubah sudut dan lain-lain.

2.8.2: FXDraw (<http://www.efofex.com>)



Rajah 2.4 Perisian FXDraw

Gambarajah di atas menunjukkan satu perisian yang dapat dimuatturun dari internet. Perisian ini dinamakan “FXDraw”. Perisian tersebut merupakan satu perisian matematik.

Ciri-ciri utama sistem:

- Dapat melukis bentuk geometri asas, mengubah saiz bentuk yang dilukis.
- Dapat melukis graf taburan normal.
- Dapat menwarnakan bentuk yang dilukis.
- Dapat melukis graf garis linear.

Keburukan:

- Terlalu kompleks dan tidak sesuai bagi kanak-kanak berumur 10-12.
- Tidak dapat mengira luas dan perimeter bentuk-bentuk geometri asas yang dilukis.
- Bukan satu perisian yang spesifik untuk membelajari geometri.

2.9 Ringkasan Bab 2

Kajian literasi adalah untuk mengkaji latarbelakang dan mengumpul maklumat untuk membangunkan “Pakej Bergrafik bagi Geometri” yang dicadangkan ini. Ia merangkumi kajian ke atas perisian yang telah sedia ada di pasaran. Ini adalah untuk mendapat pandang yang lebih jelas bagi pembangunan sesuatu sistem. Selain itu, kajian ini membolehkan perbandingan dibuat antara perisian yang sedia ada dengan perisian yang akan dibangunkan untuk mengetahui kebaikan atau kekuatan dan keburukan atau kelemahannya. Ini dapat memberikan pandangan menyeluruh tentang perisian yang akan dibangunkan dan membantu ke arah meningkatkannya. Di samping itu, kajian juga dijalankan untuk lebih memahami peranan komputer dalam pengajaran dan pembelajaran.

BAB 3 METODOLOGI

3.1 Pengenalan

Pembangunan berkesan bagi suatu projek bergantung kepada perancangan yang menyeluruh. Oleh itu, perancangan mesti disediakan bagi memandu pembangunan sistem memenuhi objektifnya. Tambahan pula, dengan membuat perancangan kita dapat menyiapkan projek pada tarikh yang ditentukan. Berikut merupakan metod yang digunakan untuk projek ini:

3.2 Metodologi Pembangunan Yang Dipilih

Metodologi yang dipilih untuk projek ini adalah pendekatan Unified.(Unified Approach) Oleh itu, proses yang digunakan adalah proses Unified. Cara yang digunakan untuk menyokong proses ini adalah UML (Unified Modeling Language). Teknik yang digunakan adalah berdasarkan pelbagai jenis diagram yang dispesifikasikan dalam UML (contohnya, Use-case Diagram, Sequence Diagram, Activity Diagram). Perisian yang digunakan untuk menyokong model-model ini adalah Rational Rose.

3.2.1 Pendekatan Unified (Unified Process)

Pendekatan Unified merupakan proses pembangunan perisian yang sangat umum dan boleh diaplikasikan kepada lingkungan perisian dan konteks yang luas. UP (Proses Unified) mempunyai beberapa ciri-ciri. Pertama, UP adalah berasaskan komponen. Ia menggunakan konsep berorientasikan objek di mana perisian adalah terdiri daripada pelbagai komponen yang berbeza. Setiap komponen berantaramuka antara satu sama

lain untuk membekalkan gol sistem tersebut. Kedua, UP adalah disokong oleh UML. UML sangat penting kepada UP disebabkan ia digunakan untuk mewakili struktur dan sifat sistem yang akan dibangunkan. Notasi UML yang berkualiti itu membantu menyalurkan maklumat secara tepat dan mengurangkan masalah salah interpretasi. Tetapi ciri-ciri UML yang paling ketara adalah berpandukan use-case, tumpuan kepada senibina dan “penokokan dan pengulangan”.

Berpandukan Use-case (Use-case Driven)

Frasa “use-case” digabungkan oleh Ivar Jacobson daripada metod kejuruteraan perisiannya yang berorientasikan objek. Use-case adalah berorientasikan pengguna kerana ia menspesifikasikan bagaimana pengguna berinteraksi dengan sistem dan ini seterusnya merangkumi semua keperluan pengguna. Oleh itu, “berpandukan use-case” (use-case driven) bermaksud kes guna merupakan satu asas kepada permulaan pembangunan sistem. Ini adalah penting dalam pembangunan sistem semasa penentuan keperluan fungsian sistem. [Ivar Jacobson, Grady Booch, James Rumbaugh, [1999], ‘*The Unified Software Development Process*’]

Tumpuan Kepada Senibina (Architecture-Centric)

Senibina sistem merujuk kepada bagaimana sistem itu diorganisasikan seperti yang didefinisikan oleh struktur, antaramuka dan sifatnya. Ia juga mengalamatkan perkara yang lebih luas seperti kebolehgunaan, pretasi dan pertimbangan bukan penstrukturan yang lain. Ini bermaksud UP sangat bergantung kepada senibina sistem untuk memandu sistem yang akan dibangunkan. Ia berkait rapat dengan kes guna. Ini penting dalam

memahami sistem yang akan dibangun, mengorganisasi pembangunan dan menggalakkan penggunaan semula komponen-komponen tertentu [Ivar Jacobson, Grady Booch, James Rumbaugh, [1999], ‘ *The Unified Software Development Process*’]

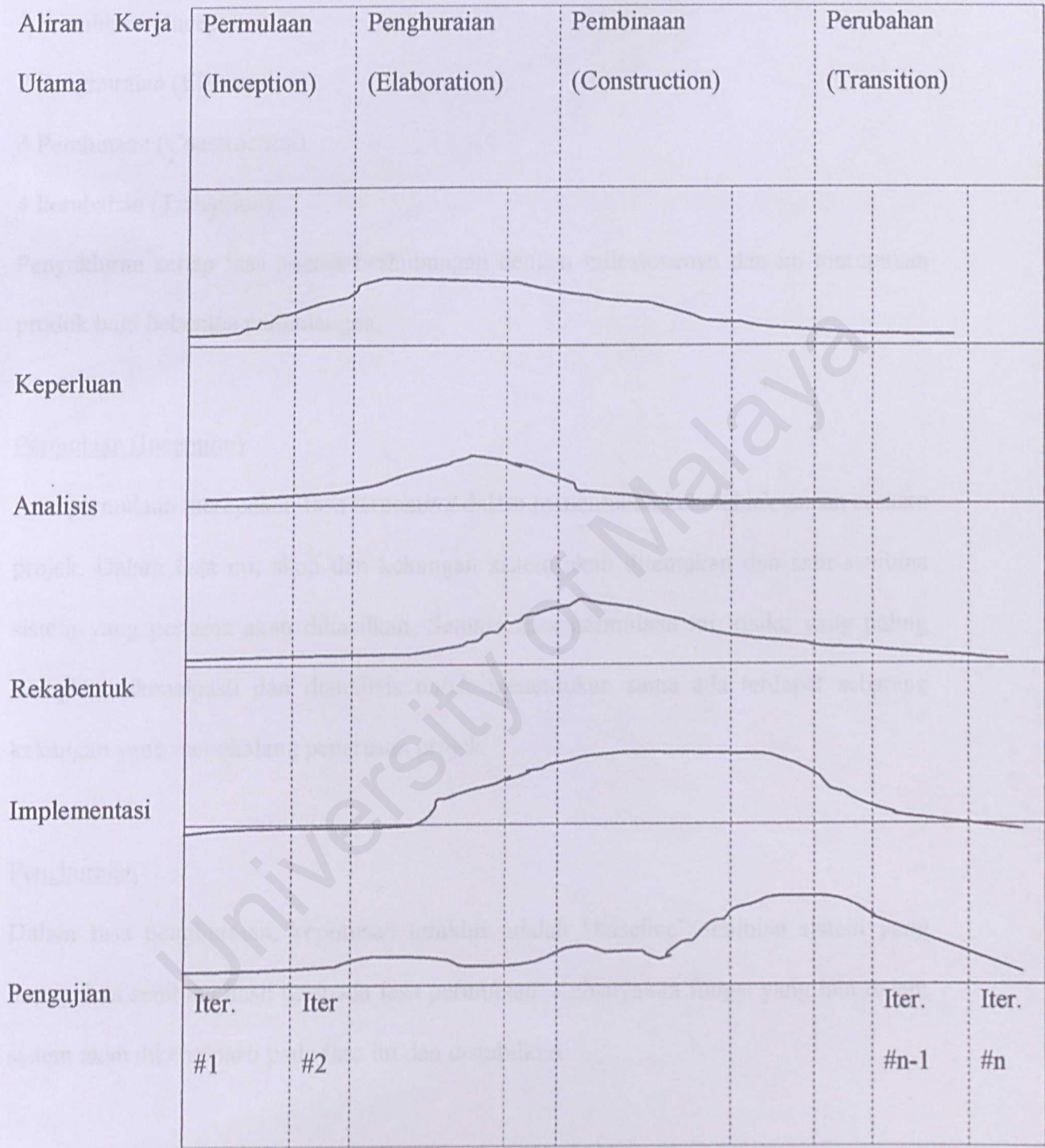
Pengulangan dan Penokokan(Interactive and Incremental)

UP menggunakan konsep pengulangan dan penokokan. Pengulangan pada dasarnya merupakan sub-projek dalam satu projek besar. Setiap pengulangan itu akan menyebabkan penokokan kepada keseluruhan projek dan mentranslasinya kepada fungsi yang lebih atau rekabentuk yang lebih teguh.

Ivar Jacobson’s mengatakan bahawa kes guna menunjukkan interaksi antara pengguna dengan sistem dan ini merupakan asas kepada analisis dan rekabentuk.

[Ivar Jacobson, Grady Booch, James Rumbaugh, [1999] ‘ *The Unified Software Development Process*’]

3.2.2 Fasa Pendekatan Unified



Rajah 3.1 Gambarajah menunjukkan fasa-fasa dan aliran kerja bagi pendekatan Unified [Ivar Jacobson, Grady Booch, James Rumbaugh, ‘The Unified Software Development Process’]

Pendekatan Unified terdiri daripada 4 fasa:

1. Permulaan (Inception)
2. Penghuraian (Elaboration)
3. Pembinaan (Construction)
4. Perubahan (Transition)

Pengakhiran setiap fasa adalah berhubungan dengan milestonenya dan ini merupakan produk bagi beberapa pengulangan.

Permulaan (Inception)

Fasa permulaan merupakan fasa terpenting dalam menentukan kebolehlaksanaan sesuatu projek. Dalam fasa ini, skop dan kekangan sistem akan ditentukan dan satu senibina sistem yang pertama akan dihasilkan. Semasa fasa permulaan ini, risiko yang paling mungkin dikenalpasti dan dianalisis untuk menentukan sama ada terdapat sebarang kekangan yang menghalang penerusan projek.

Penghuraian

Dalam fasa penghuraian, keputusan terakhir adalah “baseline” senibina sistem yang merupakan senibina hasil daripada fasa permulaan. Kebanyakan fungsi yang lain dalam sistem akan dikenalpasti pada fasa ini dan distabilkan

Pembinaan

Fasa pembinaan merupakan fasa di mana sistem akan dibangunkan secara berulang-ulang iaitu secara iteratif (iteration) dan penokokan. Baseline senibina akan dihasilkan

dan penghantaran sistem dengan kebolehoperasian yang minimal akan dipertimbangkan dalam fasa ini.

Perubahan

Dalam fasa perubahan, golnya adalah untuk menghasilkan satu sistem yang lengkap yang memenuhi keperluan-keperluan yang tertakrif. Pengujian sistem akan dijalankan pada fasa ini dan penyelenggaraan akan dijalankan untuk memperbetulkan kecacatannya.

3.3 Fasa AliranKerja (Workflow)

Dalam UP, 5 aliran kerja akan merentangi 4 fasa. Setiap aliran kerja berkaitan rapat dengan model yang menghasilkan ciri statik dan dinamik sistem. 5 aliran kerja itu adalah:

1. Keperluan
2. Analisis
3. Rekabentuk
4. Pelaksanaan
5. Pengujian

Keperluan

Aliran kerja keperluan mempertimbangkan model kes guna bagi sistem tersebut. Model ini ditunjukkan dengan menggunakan diagram kes guna. Aliran kerja keperluan adalah penting khasnya pada fasa permulaan dan fasa penghuraian di mana fungsian-fungsian yang paling kritikal akan dinyatakan dan dikenalpasti.

Analisis

Aliran kerja analisis membentuk model analisis yang “menyedarkan” (realizes) model kes guna. Model analisis membetulkan dan menstrukturkan keperluan sistem. Model analisis terutamanya penting pada fasa penghuraian.

Rekabentuk

Aliran kerja rekabentuk menghasilkan model rekabentuk bagi sistem di mana ia merupakan kesedaran fizikal bagi model use-case. Aliran kerja rekabentuk adalah penting terutamanya pada fasa penghuraian dan pembinaan.

Pelaksanaan

Tujuan aliran kerja pelaksanaan adalah menghasilkan model pelaksanaan. Model ini menerangkan bagaimana komponen-komponen sistem akan dipakejkan. Aliran kerja ini adalah penting pada fasa pembinaan.

Pengujian

Model pengujian merupakan hasil utama bagi aliran kerja pengujian. Model pengujian menerangkan pelan pengujian (test plan) dan kes pengujian (test case) yang akan dipersembahkan. Kebanyakan aliran kerja ini wujud dalam fasa pembinaan.

3.4 Unified Modeling Language(UML)

Pendekatan Unified dibangunkan secara meluas sejak 30 tahun dahulu oleh pelbagai pihak. Tetapi penyumbang utamanya adalah Ivar Jacobson, Grady Booch dan James Rumbaugh.

UML merupakan kombinasi Kejuruteraan Perisian Berorientasikan Objek oleh Ivar Jacobson, Teknik Permodelan Objek oleh Jim Rumbaugh dan Metod Booch oleh Grady Booch. UML direkabentuk untuk permodelan proses secara tidak bergantung tetapi adalah sesuai dan baik jika digunakan bersama Pendekatan Unified. [Kendal].

UML adalah bahasa grafik yang mempunyai satu set peraturan dan simantiks tertentu serta digunakan untuk melihat , menspesifikasikan, membina dan mendokumenkan artifak, iaitu output setiap fasa atau tahap pembangunan. UML juga menawarkan satu cara piawai untuk menulis blueprint atau dokumen bagi sesuatu sistem, ini termasuk bahan konseptual seperti proses perniagaan dan fungsi sistem. Terdapat sebanyak sembilan rajah berbeza yang disediakan oleh UML. Antaranya ialah rajah kes guna, rajah kelas, rajah berjujukan, rajah kolaborasi, rajah keadaan, rajah aktiviti, rajah objek, rajah komponen dan rajah penggunaan.

Beberapa gol utama bagi dalam rekabentuk UML adalah :

Membekalkan pengguna satu bahasa model yang bervisual dan sedia digunakan supaya mereka dapat membangunkan dan menukar model-model yang bermakna.

Membekalkan mekanisma untuk pengembangan dan penspesifikasian skop projek daripada konsep utama yang telah digariskan

Menggalakkan pengembangan perisian berorientasikan objek di pasaran.

Mengintegrasikan praktik dan metodologi yang baik.

[*The Unified Modeling Notation Guide Version 1.1* written by Grady Booch, Ivar Jacobson, and James Rumbaugh]

3.5 Sebab-sebab Pendekatan Unified dipilih

Sebab-sebab Pendekatan Unified dipilih adalah membolehkan peserta pembangunan sistem dapat membina model-model yang membolehkan mereka melihat sistem (visualisasi yang bererti “visualize” sistem yang akan dibangunkan supaya dapat merangkumi dan memenuhi keperluan pelanggan), menspesifikasikan struktur dan sifat sistem (Spesification-membina satu sistem yang tepat kepada keperluan sistem (precise), tidak bercanggah dengan keperluan yang dispesifikasikan (unambiguous) dan lengkap), membina sistem (menjanakan kod dari model) dan mendokumentasikan keputusan-keputusan yang dibuat di sepanjang proses pembangunan sistem.

3.6 Ringkasan Bab 3

Metodologi pembangunan sistem merupakan satu siri langkah-langkah yang harus diikuti untuk pembangunan suatu sistem. Pertimbangan harus dilakukan dalam pemilihan sesuatu metodologi supaya metodologi yang sesuai dengan pembangunan sesuatu sistem akan menjimatkan masa dan sumber. Di samping itu, kesilapan juga dapat diminimumkan.

BAB 4 ANALISIS

4.1 Pengenalan

Analisis sistem merupakan satu proses mengkaji dan menganalisis pada sistem-sistem yang sedia ada untuk memastikan keperluan, kaedah, fungsi yang harus dititikberatkan dalam sistem yang akan dibangunkan.

4.2 Teknik mendapat maklumat

Metod yang digunakan untuk mengumpul maklumat dan sumber adalah seperti berikut:

Kajian tesis dan rujukan di perpustakaan

Kajian dijalankan dengan membuat rujukan tesis yang mempunyai tajuk yang hampir sama untuk mendapat idea dalam sistem yang akan dibangunkan. Selain itu, kebaikan dan kekangan projek juga dianalisis.

Melayari internet

Banyak sumber terdapat di dalam internet. Ini termasuk perisian-perisian yang sedia ada yang boleh dimuatturunkan ke dalam komputer. Adalah satu perkara yang baik dengan menghasilkan satu perisian yang baik dengan membuat kajian terhadap sistem yang sedia ada. Membelajari dan membuat kajian bagi sistem yang sedia ada adalah baik kerana kita dapat menambahkan ciri-ciri tertentu dalam perisian tersebut dan seterusnya menghasilkan perisian yang lebih berkualiti.

Perbincangan dengan penyelia

Perbincangan dengan penyelia mengenai keperluan fungsian yang dicadangkan untuk sistem ini. Ini adalah supaya dapat membangunkan satu sistem yang memenuhi keperluan dan kesesuaian terhadap sasaran pengguna.

Soal-selidik

Kajian soal-selidik telah dijalankan dengan murid-murid dan guru sekolah rendah bagi mendapatkan maklumat mengenai pandangan mereka mengenai pengetahuan dan penggunaan komputer dalam membantu belajaran dan pembelajaran.

4.3 Keperluan Fungsian

Keperluan fungsian merupakan fungsi atau ciri-ciri yang mesti dimasukkan ke dalam sistem untuk memenuhi keperluan dan kesesuaian pengguna. Keperluan fungsian menyatakan bahawa apa yang sistem harus buat, bagaimana sistem dapat memberikan tindak balas bagi input-input tertentu dan bagaimana ia beroperasi dalam situasi-situasi tertentu. Dalam sesetengah kes tertentu, ia juga menyatakan apakah yang sistem tidak harus buat. Tambahan pula, keperluan fungsian tidak bergantung kepada pelaksanaan bagi sesuatu "solution".

Terdapat satu modul telah dikenalpasti dan dicadangkan dalam sistem ini, beberapa keperluan fungsian yang dicadangkan untuk memberikan ciri-ciri tersebut kepada sistem tersebut seperti yang di bawah:

'Browse' nota Ringkas

Keperluan fungsian ini membekalkan satu ringkasan tentang bentuk-bentuk geometri asas dalam bentuk dua dimensi (segitiga tiga sama sisi, segitiga tiga sama kaki, segitiga tak sama sisi, segiempat sama, segiempat tepat, segitiga bersudut tegak) dan tiga dimensi (kubus, kuboid, prisma, piramid bertapak segi empat sama, piramid bertapak segi empat tepat, silinder, kon dan sfera). Di samping itu, dalam nota-nota ringkas tersebut juga dibekalkan formula pengiraan luas dan perimeter bagi segiempat sama, segiempat tepat dan segitiga manakala pengiraan isipadu pula hanya ditunjukkan untuk kubus dan kuboid sahaja yang mengikut kandungan kursus KBSR Matematik tahun 4, 5 dan 6.

Melukis

Peralatan melukis merupakan keperluan fungsian yang dicadangkan di mana ia dapat membekalkan kanak-kanak satu "tool" untuk menghasilkan lukisan bentuk-bentuk geometri asas

Mengubahsuai ciri-ciri bagi bentuk yang telah dilukis

Selain melukis, pengguna dapat

- Menwarnakan lukisan yang diingini
- Menggerakan bentuk yang telah dilukis
- Memusingkan bentuk yang telah dilukis
- Memilih jenis warn yang diingini bagi bentuk yang ingin dilukis

Mengubahsuai ciri-ciri segitiga

Satu skrin mengenai segitiga khasnya telah dibekalkan untuk :

- Mengubah kecondongan sisi segitiga
- Mengubahsuai skala sisi segitiga
- Memusingkan segitiga.

Ini dicadangkan kerana terdapat beberapa jenis segitiga yang berbeza dan dengan membekalkan skrin diharapkan dapat membantu para pelajar sekolah rendah mengenali jenis-jenis sigitiga.

Melukiskan polygon

Polygon dilukis dengan

- Menginput nilai x
- Menginput nilai y

Selepas nilai-nilai tersebut diinput, bentuk lukisan polygon dapat dipaparkan.

Melukis teks

Pengguna dapat 'melukis' teks dengan memilih jenis font dan saiz yang dikehendaki.

Mengira perimeter dan luas bentuk

Luas dan perimeter yang telah dilukiskan dapat ditentukan di atas satu status bar supaya pengguna dapat mengetahui luas dan perimeter bentuk yang telah dilukis.

Paparan koordinat x dan y

Diharapkan perisian yang akan dibangunkan ini dapat memaparkan kedudukan tertentu yang telah dilukis dengan pengukuran yang tertentu.

Keluar (Exit)

Membolehkan pengguna untuk keluar daripada sistem jika mereka ingin meninggalkan perisian ini.

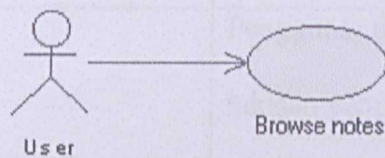
4.3.1 Rajah Kes Guna (*Use Case Diagram*)

Rajah ini memaparkan secara bergrafik di antara sistem dengan sistem luaran dan pengguna (*actor*). Ia turut menjelaskan siapakah yang akan menggunakan sistem dan bagaimanakah kemahuan pengguna untuk berinteraksi dengan sistem.

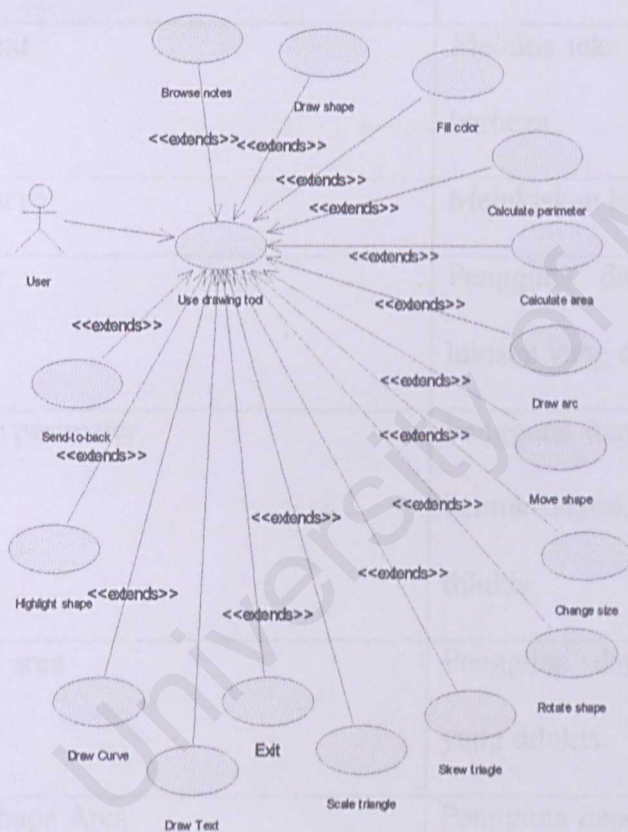
Kes guna (*Use Case*) diberikan nama mengikut pandangan yang berkaitan dengan pengguna sistem, contohnya "Browse notes".

Aktor (*actor*) pula merupakan satu set koheren yang berinteraksi dengan kes guna. Ia boleh terdiri daripada pengguna (manusia), perkakasan dan sistem yang lain yang berinteraksi dengan kes guna. Aktor diwakili dengan manusia lidi.

Gambarajah berikut menunjukkan satu contoh pengguna berinteraksi dengan kes guna yang dikenali "Browse notes". Anak panah tersebut menunjukkan pengguna "initiates" kes guna tersebut.



Rajah 4.1 Contoh kes guna



Rajah 4.2 Gambarajah Kes Guna bagi sistem

Nama Kes Guna	Penerangan
Browse notes	Pengguna boleh membaca nota ringkas

	yang dibekalkan.
Use drawing tool	Pengguna dapat menggunakan aplikasi lukisan yang dibekalkan.
Exit	Pengguna dapat keluar dari sistem jika pengguna ingin meninggalkan sistem.
Draw shape	Pengguna dapat melukis bentuk-bentuk geometri asas.
Draw Text	Melukis teks dengan font dan saiz yang berbeza.
Draw Curve	Melukiskan lengkung.
Fill color	Pengguna dapat mengisi warna bagi lukisan yang dihasilkan.
Calculate perimeter	Pengguna dapat mengira perimeter bagi bentuk-bentuk geometry asas yang dilukis.
Calculate area	Pengguna dapat mengira luas bentuk yang dilukis.
Change shape Arcs	Pengguna dapat mengubah sudut bentuk yang dilukis.
Move shape	Pengguna dapat menggerakkan bentuk yang dilukis.
Change size	Pengguna dapat mengubah saiz bentuk yang dilukis.

Rotate shape	Memusingkan bentuk yang dilukis.
Skew triangle	Memesongkan sisi segitiga
Scale triangle	Penskalaan segitiga iaitu pembesaran atau pengecilan

Jadual 4.1 :Penerangan tentang nama kes guna

Aktor	Penerangan
User	Pengguna bagi sistem tersebut.

Jadual 4.2: Penerangan aktor dalam sistem

4.4 Keperluan Bukan Fungsian

Keperluan bukan fungsian merupakan di mana sistem mesti beroperasi atau piawaian yang harus ada pada sistem yang terhasil. Ia dirumuskan seperti berikut:

Kebolehpercayaan dan keberkesanan

Semua kesalahan dan kegagalan yang mungkin akan dipertimbangkan semasa dalam langkah pengujian. Setiap fungsi mesti beroperasi seperti yang ditakrifkan. Contohnya, fungsi “melukis” harus beroperasi untuk melukis sesuatu bentuk tetapi bukan mewarna.

Antaramuka yang interaktif dan senang difahami oleh pengguna

Sistem ini akan berasaskan konsep mesra pengguna dan demi mencapai tujuan ini, antaramuka mesti menarik dan ramah mesra.

Masa maklum balas

Semua maklumat yang diinginkan oleh pengguna harus dioutputkan secepat mungkin kepada pengguna. Sistem ini harus menjimatkan masa pengguna.

Kebolehselenggaraan

Sistem ini direka agar kerja-kerja yang diperlukan untuk selenggara, menentukan kesalahan adalah minimal. Komen atau penerangan yang sesuai diletakkan pada kod-kod program supaya system ini mudah diselenggara.

Keteguhan

Sistem harus berupaya mengendalikan situasi yang tidak normal dengan berkesan, berupaya untuk meneruskan perjalanan sistem walaupun terdapat gangguan-gangguan yang kecil.

4.5 Pertimbangan Sistem Pengendalian

4.5.1 Pengenalan

Sistem pengendalian ialah sistem perisian yang mengawal dan mengurus semua aktiviti dalam komputer. Secara amnya, satu sistem pengendalian mempunyai tiga fungsi utama, iaitu :

- mengumpulkan dan membahagi sumber sistem, ia membekalkan lokasi dalam ingatan utama untuk data dan program, dan mengawal peranti input dan output
- menjadualkan penggunaan sumber komputer dan kerja komputer
- memantau aktiviti sistem komputer, sentiasa mengawal setiap kerja komputer dan mengesan siapa yang sedang menggunakan sistem tersebut

4.5.2 Windows 98

Windows 98 direkabentuk untuk pasaran komputer peribadi dan merupakan sistem pengendalian berantaramuka pengguna grafik. Ini adalah cara yang membenarkan pengguna berinteraksi dengan komputer yang menggunakan gambar atau grafik. Windows 98 ini akan dilarikan pada pemproses berasaskan Intel sahaja dan tidak akan menyokong motherboard yang dilengkapi dengan multipemproses.

Walau bagaimanapun, sistem pengendalian ini akan menyokong nama fail yang panjang dan mempunyai keupayaan rangkaian yang terbina di dalam termasuklah penyambungan ke Windows NT Server atau stesen kerja. Windows 98 bukanlah 32 bit yang lengkap maka pembangun telah menggunakan gabungan kod 32 bit dan 16 bit untuk membekalkan keupayaan yang maksimum di antara aplikasi baru dan lama.

Pengguna bebas daripada keperluan penyelenggaraan jika menggunakan Windows 98 dan ini memberi ciri-ciri yang berkuasa kepada Windows 98 kerana Windows 98 adalah lebih mudah dibangun, disokong, diselenggara, dikesan masalah dan juga diuruskan.

Windows 98 menyokong perkakasan dalam generasi baru seperti multimedia untuk pelajaran dan pengajaran dengan sepenuhnya. Masalah keserasian akan diselesaikan dan penambahan perkakasan luaran seperti pencetak menjadi lebih mudah dengan Universal Serials Bus (USB).

4.5.3 Windows NT 4.0

Windows NT 4.0 direkabentuk untuk menghasilkan sebuah sistem pengendalian yang berintegrasi ,komprehensif dan ia juga merupakan sistem pengendalian pelayan yang mudah.Ianya mempunyai ciri-ciri yang baik dari segi keselamatan ,kemudahalihan dan kebolehsesuaian . Ia juga merupakan penyelesaian lengkap untuk urusan perkhidmatan dalam intranet dan internet .

Windows NT 4.0 menyokong teknologi yang dipanggil Redundant Array of Inexpensive Disks (RAID) . RAID memberi perlindungan kepada data dan ini membantu pelayan web daripada crash dengan mudah yang disebabkan oleh kegagalan cakera keras.Integrasi yang baik antara Internet Information Server (IIS) dengan Windows NT 4.0 menyediakan platform yang amat baik untuk pelbagai perkhidmatan dan aplikasi dalam julat yang besar. Maka ini menjadikannya sebuah sistem pengendalian yang baik.Ia juga mampu untuk memudahkan kerja yang berkaitan dengan pengurusan persekitaran rangkaian.

4.5.4 Windows 2000

Windows 2000 telah direkabentuk oleh pihak Microsoft sebagai sebuah platform yang menyokong kepelbagaian penggunaan (Sistem pengendalian pelbagai guna). Ia sebuah platform yang amat baik bagi tujuan pembangunan aplikasi berasaskan web. Selain itu , ia juga merupakan sebuah sistem pengendalian yang teguh dan selamat.Windows 2000 adalah merupakan sistem pengendalian 32 bits dan menyokong senibina cip Intel IA-64.

Sistem pengendalian Windows 2000 adalah hampir sama dengan Windows NT 4.0, cuma ia merupakan satu kemajuan yang besar ke atas Windows NT 4.0 dari segi persembahan,

kebolehpercayaan, keselamatan, kemudahan pentadbiran sistem serta antaramuka grafik pengguna telah diperbaiki dan dikemaskinikan bagi menghasilkan sistem pengendalian yang amat baik.

Secara amnya, Windows 2000 membekalkan perkhidmatan yang disenarai di bawah:

- Perkhidmatan aplikasi web seperti Internet Information Server 5 dan ASP Programming Environment
- Perkhidmatan Direktori
- Perkhidmatan keselamatan seperti Kerberos Authentication
- Perkhidmatan rangkaian dan komunikasi seperti Internet Connection Sharing
- Perkhidmatan Pengurusan seperti Management Console
- Perkhidmatan terminal
- Perkhidmatan cetak dan fail seperti Distributed File System

4.5.5 UNIX

Unix adalah merupakan satu sistem pengendalian yang popular disebabkan sokongnya yang besar terhadap tapak (bases) dan agihan (distrubution) . Pada mulanya ia dibina sebagai sistem pelbagai tugas (multitasking system) untuk mini komputer dan mainframe pada pertengahan 1970-an. Tetapi kini ,UNIX telah menjadi salah satu sistem pengendalian yang popular di seluruh dunia walaupun terdapat kelemahan padanya seperti antaramuka yang mengelirukan .

UNIX merupakan sistem pengendalian pelbagai pengguna yang membolehkan sebilangan pengguna menggunakan satu komputer pada suatu masa dan melarikan pelbagai aplikasi yang

berlainan. Dalam UNIX pengguna perlu login terlebih dahulu sebelum dapat menggunakan sistem tersebut.

4.6 Pemilihan Sistem Pengendalian

Sistem pengendalian yang dipilih untuk membangunkan perisian ini adalah Windows 2000 Professional Edition

Sebab-sebab pemilihan Windows 2000 Professional Edition:

- Antaramuka yang mesra-pengguna disebabkan sasaran pengguna adalah pelajar sekolah rendah.
- OS ini stabil dan sesuai untuk penggunaan sistem yang tanpa melibatkan web atau rangkaian.
- Penyelenggaraannya juga mudah jika terdapat sebarang gangguan.

4.7 Pertimbangan Bahasa Pengaturcaraan

4.7.1 Pengenalan

Jenis bahasa yang dipilih tidak semestilah perlu terlalu hebat atau kompleks sebaliknya apa yang penting di sini adalah memberi kepuasan kepada pengguna dari segi masa tindak balas, keupayaan menarik pengguna serta mudah difahami . Antara ciri-ciri yang perlu dipertimbangkan dan memilih bahasa pengaturcaraan yang sesuai adalah seperti berikut:

- Bahasa pengaturcaraan mestilah berupaya untuk menyokong komunikasi keperluan-keperluan fungsian tertentu.

- Bahasa pengaturcaraan yang memberi kemudahan untuk rekabentuk antaramuka pengguna yang bercirikan grafik. Antaramuka ini selalu digunakan dalam sistem yang berinteraksi dengan pengguna akhir.
- Kelebihan atau ciri-ciri unik yang terdapat pada bahasa pengaturcaraan tersebut.

4.7.2 Visual Basic 6.0

Visual Basic merupakan satu bahasa pengaturcaraan konvensional yang digemari oleh pembangun sistem kerana mempunyai ciri-ciri yang menarik. Bahasa ini berasaskan antaramuka pengguna bergrafik. Visual Basic juga merupakan satu cara yang mudah untuk menulis program untuk Windows.

Setiap fungsi yang dilakukan oleh objek akan dikodkan dengan cepat kerana penekannya hanya diberikan kepada fungsi yang akan dilakukan oleh objek tersebut. Tambahan pula, masa untuk merekabentuk objek dapat dikurangkan dengan menggunakan bahasa pengaturcaraan ini.

Program yang dibangunkan dengan bahasa pengaturcaraan ini amat sesuai dengan Windows 95 atau lebih tinggi. Selain itu, Visual Basic juga dapat menyokong pelbagai jenis pangkalan data seperti Microsoft Access, Informix, DBASE.

Terdapat pelbagai fungsi dalaman seperti 'Object Linking and Embedding (OLE)' dan 'Dynamic Data Exchange (DDE)' yang dapat membantu dalam pembangunan sesuatu sistem.

Visual Basic juga menyokong 'ODBC' (Object Database Connection) yang membolehkan capaian kepada pelayan dan pangkalan data tempatan termasuk Microsoft SQL Server, Oracle dan persekitaran pelayan-pelanggan.

4.7.3 Java

Nama Java merupakan tanda perniagaan (trademark) bagi Sun Microsystems dan merujuk kepada bahasa pengaturcaraan yang dibangunkan oleh Sun pada 23 Mei 1995. Secara umum, Java adalah persekitaran yang mudah, berorientasikan objek, dinamik, mempunyai prestasi tinggi dan selamat .

Pengaturcaraan berorientasikan objek dalam Java merupakan satu cara yang sangat berkuasa dalam membangunkan perisian. Dalam pengaturcaraan berorientasikan objek, program komputer boleh dipertimbangkan sebagai satu kumpulan objek yang boleh berinteraksi antara satu dengan yang lain.

Ciri yang paling baik dalam Java adalah ia boleh digunakan untuk menghasilkan program yang boleh dilaksanakan dalam halaman World Wide Web. Program ini dikenali sebagai applets. Applet Java yang berada dalam rangkaian membolehkan pengguna hanya “download” fungsian perisian dan data yang diperlukan untuk menjalankan tugas. Pengguna tidak perlu menyelenggara aturcara perisian atau fail data yang besar dalam mesin komputer mereka. Apabila pengguna selesai dalam pemprosesan, data boleh disimpan melalui rangkaian. Java boleh digunakan oleh komputer rangkaian kerana Java membenarkan semua pemprosesan data dan perisian boleh distorkan dalam pelayan rangkaian dan boleh “download” melalui rangkaian apabila diperlukan.

Untuk melarikan suatu perisian Java, satu komputer memerlukan suatu sistem pengendalian yang mengandungi JVM (Java Virtual Machine). JVM dilaksanakan dalam perisian atas mesin sebenar dan menspesifikasikan set arahan, set pendaftar, format kelas fail, stack, medan ingatan.

Walaupun program berdasarkan Web adalah kelebihan bahasa ini, tetapi Java juga merupakan satu bahasa pengaturcaraan multi-tujuan yang digunakan untuk pembangunan pelbagai jenis program.

4.7.4 MatLab

Matlab bermaksud MATrix LABoratory yang dibangunkan oleh The Mathworks, Inc.

Matlab merupakan satu sistem perisian yang interaktif bagi komputasi numerikal, visualisasi saintifik, komputasi simbolik dan grafik. Seperti yang dicadangkan oleh namanya, Matlab direkabentuk khas untuk komputasi matriks seperti menyelesaikan masalah “linear equations”, “computing eigenvalues” dan “eigenvectors”, pemfaktoran matriks dan sebagainya. Tambahan pula, Matlab mempunyai pelbagai kebolehan untuk bergrafik, dan boleh dilanjutkan (extended) melalui program yang ditulis dalam bahasa pengaturcaraannya sendiri. Banyak program ini datang bersama dengan sistem; beberapa kebolehan lanjutan bagi Matlab kepada masalah bukan linear, contohnya penyelesaian kepada masalah dengan nilai terdahulu bagi “equation” berlainan yang normal.

Matlab direkabentuk untuk menyelesaikan masalah numerikal, iaitu suatu aritmetik yang terhingga. Oleh itu, ia menghasilkan nilai jangkaan berbanding dengan jawapan yang tepat, dan tidak boleh dikelirukan dengan sistem komputasi yang simbolik (Symbolic Computation System SCS) seperti Mathematica atau Maple. Harus difahamkan supaya ini tidak menjadikan Matlab lebih baik atau kurang baik daripada SCS kerana Matlab merupakan suatu peralatan untuk kerja-kerja berlainan dan ini tidak boleh dibandingkan secara terus.

Antara ciri-ciri Matlab adalah:

- Perlahan berbanding dengan FORTRAN dan C kerana ia merupakan bahasa interpreted, contohnya tidak pre-kompil.

- For loops dielakkan dan menggunakan form vector (vector form).
- Pengurusan memori secara automatik , contohnya tidak perlu mengisytiharkan arrays.
- Intuitive, senang digunakan.
- Kompact (menangani array menyerupai fortran90)
- Masa pembangunan program yang pendek berbanding dengan bahasa pengaturcaraan lain seperti Fortran dan C
- Boleh menukarkan kodnya kepada kod C melalui pengkompil Matlab untuk keberkesanan yang lebih tinggi.

- Banyak toolbox yang spesifikasi aplikasinya didapati.
- Berpasang dengan Maple untuk komputasi simbolik.

4.8 Perisian Pembangunan Yang Dipilih

Java dipilih untuk membangun sistem ini:

Java adalah berorientasikan objek. Java objek terdiri daripada sekyen-sekyen kod “stand-alone” yang direkabentuk khasnya bagi sesuatu input yang tertentu menghasilkan output yang diinginkan. Ini memudahkan penambahan objek baru kepada bahasa Java dan meningkatkan kebolehannya. Java 2 juga menambahkan ciri-ciri pakejnya dalam melukis bentuk 2 dimensi dan Java 3D untuk melukis bentuk 3 dimensi. Kebolehan ini menyebabkan Java sangat berguna dalam pembangunan perisian bergrafik. Di samping itu, Java juga mempunyai ciri-ciri multimedia yang sesuai untuk membangun aplikasi bergrafik.

Contohnya, dalam JFC (Java Foundation Class) terdapat 5 komponen yang sangat berguna untuk membangun GUI(Graphic user interface) dan grafik. Java Foundation Class merupakan salah satu daripada ciri Java.

Java API (Aplication Programming Interface) pakej yang terdapat dalam JFC yang digunakan untuk pembangunan sistem:

- •Java.awt-mengandungi kelas dan antaramuka yang diperlukan untuk GUI tanpa bergantung kepada jenis platform.
- •Java.awt.dnd-menyokong operasi drag-and-drop
- •Java.awt.event-pengendalian peristiwa
- •Java.awt.geom-membekalkan kelas java 2D untuk menakrif dan mempersembahkan objek 2-dimensi dan ruangnya.

- Java.awt.image-mengandung kelas dan antarmuka yang dapat menstor dan memanipulasikan imej dalam program.
- Javax.swing-mengandung kelas dan antarmuka untuk menyokong Swing GUI
- Java.awt.color-membekalkan kelas java 2D dengan warna.

Dengan berbekalkan API ini, JAVA sudah dapat membangunkan satu aplikasi atau 'applet' yang lengkap dan dapat berfungsi dengan baik dengan menggunakan pakej-pakej ini. Di samping itu, JAVA juga mempunyai ciri 'platform independent' iaitu ia tidak bergantung kepada jenis Sistem Pengoperasian untuk membolehkannya melaksanakan fungsinya. Maka ia amat sesuai digunakan untuk pembangunan perisian kerana perisian yang dibangunkan itu dapat menyokong pelbagai jenis platform iaitu Microsoft Windows, Linux, Macintosh dan sebagainya.

4.9 Kebaikan Java

Java adalah bahasa pengaturcaraan yang **mudah**. Mudah di sini bererti ciri-cirinya penting diterangkan dengan ringkas dan padat. Java direkabentuk sedemikian supaya memudahkan pembelajaran oleh para pengaturcara.

Java adalah **berorientasikan objek**. Perisian yang berorientasikan objek mempunyai kebaikan dari segi ekonomi iaitu kebolegunaan semula komponen perisian. Di samping itu, penyelenggaraan dan penambahbaikan sistem yang besar juga mudah dikendalikan.

Java adalah **teragih**. Java dilengkapi dengan librari untuk aplikasi rangkaian menjadikannya mudah untuk pengaturcara mengurus dengan protokol paras rendah seperti TCP/IP (Transport

Control Protocol and Internet Protocol) dan protokol paras tinggi seperti HTTP (Hyper Text Transfer Protocol) dan FTP (File Transfer Protocol).

Java adalah **interpreted**. Dengan fail sumber kod “file.java” sebagai input, kompiler Java akan menjana kod bit paras rendah yang dikenali “file.class”. Fail kod bit akan dihantar kepada “interpreter” java untuk pelaksanaan. Kompiler Java dikenali sebagai javac, dan “interpreter” Java dikenali sebagai Java. Langkah-langkah untuk mengkompil dan melarikan program Java adalah : pertama, kod sumber dikompilkan dalam bentuk fail objek, kemudian fail objek akan dihubungkan bersama dengan librari luaran (external library) dan satu program yang boleh terlaksana akan dijana. Akhirnya, program adalah dilarikan dan terlaksana. Maka Java adalah satu bahasa pengaturcara yang baik untuk “rapid prototyping”. Bytecode interpreter, yang dikenali Java Virtual Machine, simulasi satu chip yang memahami Java. Virtual Machine telah dibangunkan bersesuaian pelbagai platform.

Java adalah **utuh (robust)**. Java adalah “strong-typed”, iaitu tidak perlu membuat pengisytiharan secara implicit seperti C dan jenis data (data type) tidak boleh dicampurkan. Kebaikan ini membolehkan compiler menyemak kesalahan yang mungkin timbul disebabkan oleh perpadanan jenis data yang salah. Oleh itu, ini dapat melindungi pengaturcara daripada kesalahan mereka sendiri.

Java adalah **selamat (secure)** di mana Java direkabentuk dalam “network environment”, bermaksud aplikasi dapat dimuatturun dari Internet dan dijalankan dalam mesin anda. Java akan memastikan aplikasi yang dimuatturunkan tidak melampaui kawalan Java dengan menggunakan “code verification”. Kawalan keselamatan yang kedua adalah menghalang program daripada tulis ruang disk.

Java adalah **multithreaded**. Bahasa Java menyokong multithreading di mana librari Java disertai dengan cara primitif untuk memulakan satu thread, melarikan satu thread dan menghentikan satu thread dan mendapatkan status untuk thread yang sedang dilarikan. Librari juga menawarkan satu set keserentakan primitif, di mana ia berasaskan monitor dan paradigm keadaan.

Berdasarkan kebaikan-kebaikan yang dinyatakan di atas, kebanyakannya berbanding dengan bahasa pengaturcaraan lain juga menyebabkan JAVA dipilih untuk pembangunan perisian ini.

4.10 Perkakasan minimum yang diperlukan

- Komputer dengan memori sekurang-kurangnya 64MB
- Processor sekurang-kurangnya pentium

4.11 Perisian yang dipilih

- Java2 SDK Standard Edition v1.3.1
- Java2 Runtime Environment

4.12 Perisian integrasi:

- Notepad
- Ms-Dos Prompt
- JCreator LE 2.0

4.13 Ringkasan Bab 4

Keperluan sistem ditakrifkan sebagai satu ciri atau huraian mengenai apa yang sepatutnya dilakukan oleh sistem untuk memenuhi objektif. Dalam proses pembangunan perisian, salah satu aktiviti yang terlibat ialah mendapat maklumat mengenai keperluan sistem. Analisis keperluan ini adalah bertujuan untuk memahami keperluan ini dengan terperinci dan dapat mengenalpasti apakah ciri-ciri dan maklumat yang perlu ada pada pakej. Di samping itu, melalui analisis kita juga dapat mengenalpasti apakah perisian dan perkakasan yang sesuai digunakan untuk membangunkan sistem tersebut.

5.2 Rajah Kelas (Class Diagram)

Rajah ini menggambarkan kelas-kelas dalam sistem dan objek-objek tersebut, serta nilai-nilai sebagai atribut-atribut. Ia menggambarkan hubungan-hubungan yang ada antara sistem-sistem tersebut untuk mengenalpasti sistem perisian yang sesuai untuk membangunkan sistem tersebut. Contoh: Rajah 5.2 menunjukkan kelas-kelas dalam sistem.

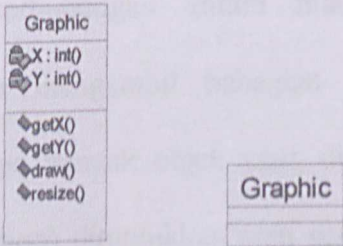
BAB 5 REKABENTUK SISTEM

5.1 Pengenalan

Fasa ini memerlukan pembangun sistem menukar maklumat dan data yang diperolehi dari fasa analisa kepada maklumat yang boleh difahami oleh pengguna. Rekabentuk skrin yang dilakukan secara lakaran (mork up) membolehkan rekabentuk skrin sebenar mudah dilakukan dan pembangun tidak akan hilang punca. Walau bagaimanapun, rekabentuk ini mungkin berubah dari semasa ke semasa bergantung kepada kreativiti serta daya imaginasi pembangun sistem bagi memastikan ianya memenuhi kehendak pengguna. Pemilihan jenis teks, latarbelakang dan grafik yang sesuai turut dipertimbangkan dengan tujuan untuk menghasilkan skrin yang menarik.

5.2 Rajah Kelas (Class Diagram)

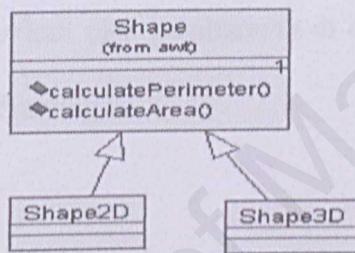
Rajah ini memaparkan kelakuan atau keseketikaan objek sebenar, iaitu nilai sebenar bagi atribut-atribut. Ia menyediakan mekanisme untuk pembangun sistem maklumat untuk mendapat suatu gambaran pantas mengenai objek-objek bagi sistem pada satu masa yang tertentu. Contoh kelas adalah dilukiskan seperti berikut.



Rajah 5.1 Contoh-contoh kelas

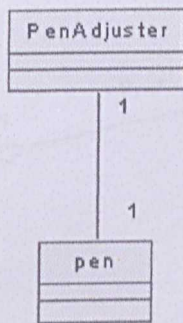
Kelas-kelas mempunyai hubungan antara mereka. Di antara hubungan yang ada dalam diagram kelas adalah generalisasi (Generalization), perhubungan (association) dan aggregasi (aggregation).

Generalisasi merupakan suatu hubungan antara satu kelas super dengan kelas yang lebih spesifik (subkelas). Ia juga dikenali sebagai hubungan “is a kind of” di mana subkelas adalah “is a kind of” superkelas. Satu contoh generalisasi adalah Shape 2D dan Shape 3D diwarisi dari kelas Shape.



Rajah 5.2 Contoh kelas yang menunjukkan Generalisasi

Satu perhubungan (association) merupakan satu hubungan berstruktur yang menspesifikasikan hubungan antara dua kelas. Satu perhubungan boleh mempunyai nama(name), peranan(role) dan multipliciti. Nama merupakan satu kata nama yang digunakan untuk menerangkan perhubungan antara dua kelas. Peranan pula menerangkan bagaimana kelas itu mengambil bahagian dalam satu hubungan. Multipliciti pula menunjukkan berapa banyak objek yang dihubungkan dengan satu objek. Contoh perhubungan (association) ditunjukkan oleh gambarajah di bawah yang menunjukkan satu “pen adjuster” mempunyai satu “pen”.



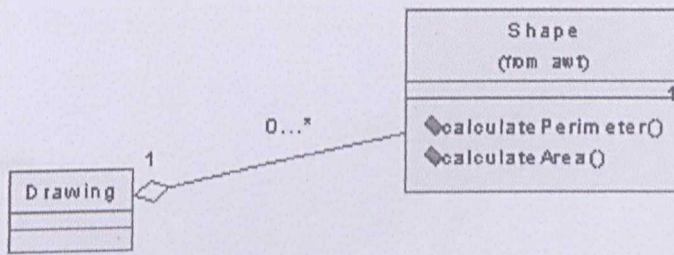
Rajah 5.3 Contoh kelas menunjukkan Perhubungan (Association)

Hubungan multicipliciti pula ditunjukkan oleh gambarajah di bawah dengan satu kelas “pen” boleh menghasilkan banyak “drawing”.



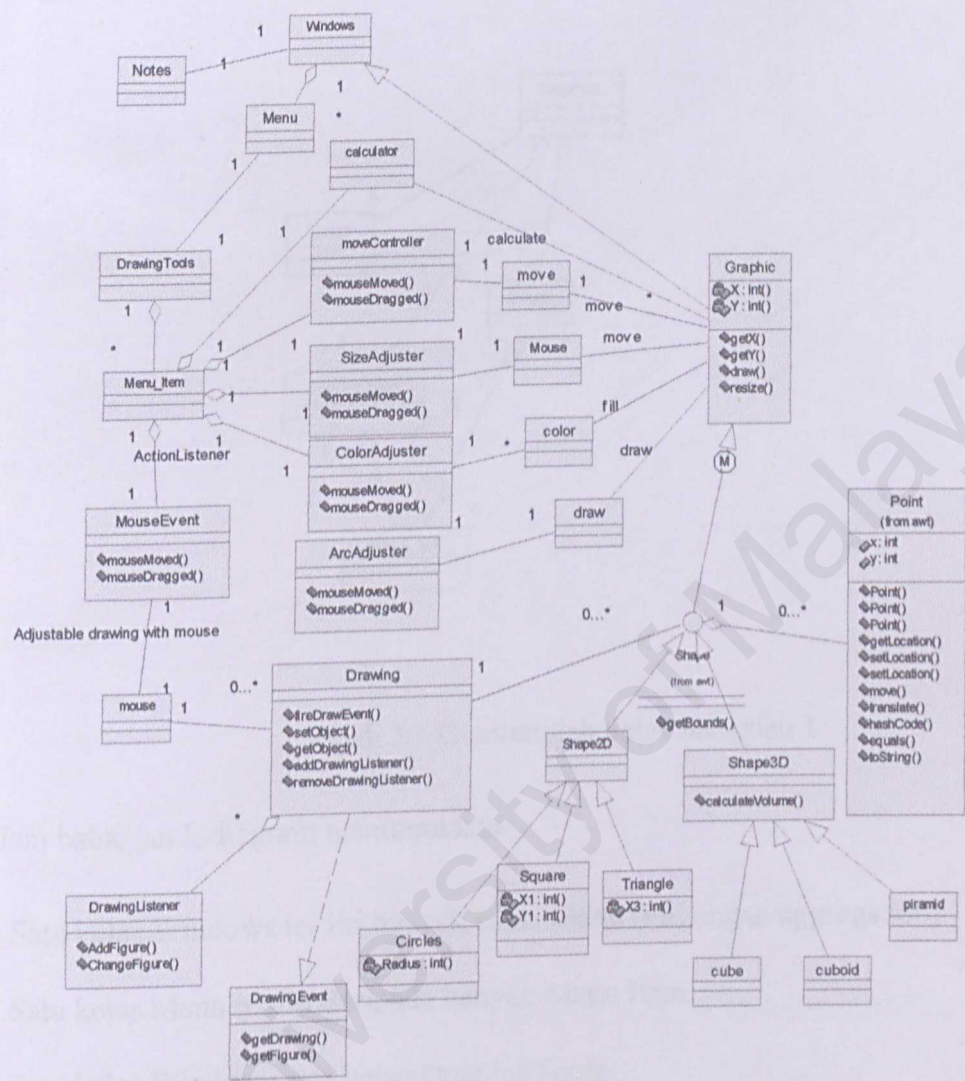
Rajah 5.4 Contoh kelas menunjukkan Perhubungan (Association) satu kepada banyak

Aggregasi merupakan satu hubungan istimewa. Ia menghubungkan satu kelas kepada kelas yang lain dan memodelkan hubungan “whole/part” atau “has-a” dan membawa maksud satu kelas merupakan sebahagian daripada kelas yang lain. Contohnya ditunjukkan di bawah bahawa satu kelas “drawing” terdiri daripada 0 hingga banyak “Shape”.



Rajah 5.5 Contoh kelas menunjukkan hubungan Pengaggregatan(Aggregation)

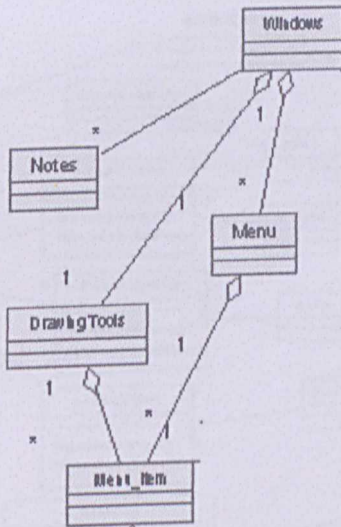
Gambarajah Kelas.



Rajah 5.6 Gambarajah kelas bagi sistem

Gambarajah kelas akan disemak bahagian demi bahagian.

Bahagian 1

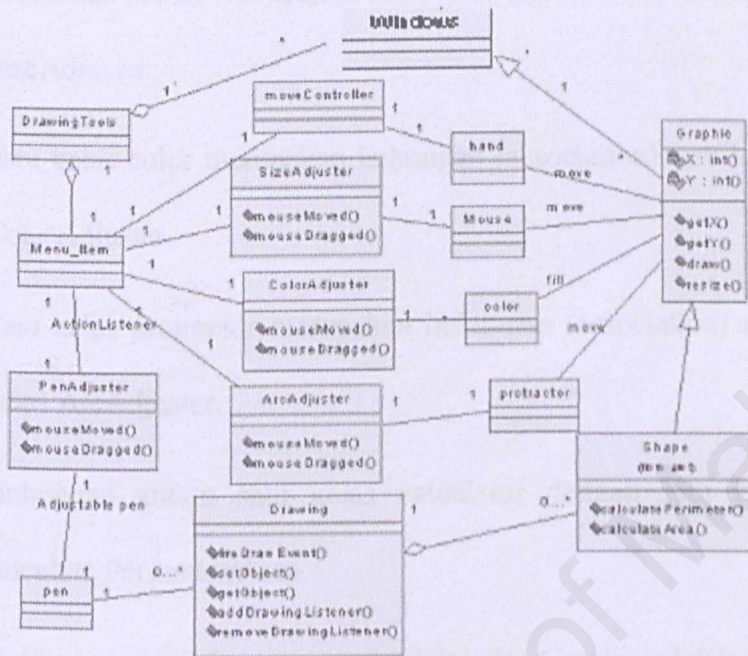


Rajah 5.7 Gambarajah kelas bahagian 1

Dalam bahagian I, diagram menunjukkan:

- Satu kelas Windows terdiri banyak kelas Menu (hubungan aggregation).
- Satu kelas Menu terdiri daripada banyak Menu Item.
- Satu kelas Windows satu kelas DrawingTools.
- Satu kelas Windows terdiri daripada banyak kelas Notes

Bahagian II



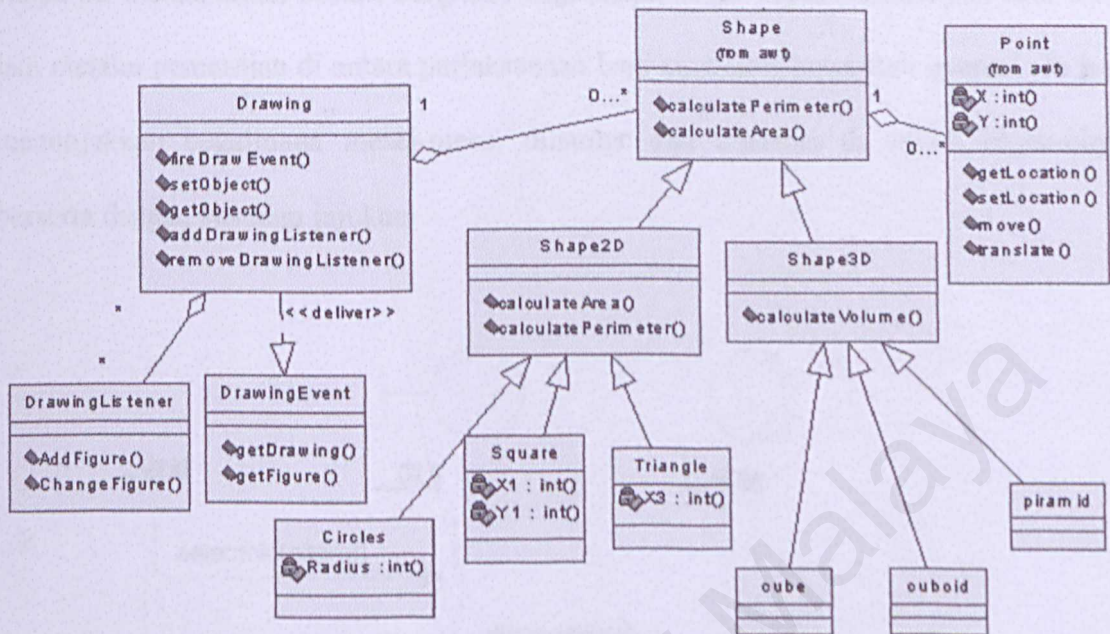
Rajah 5.8 Gambarajah kelas bahagian II

Dalam bahagian II, diagram menunjukkan:

- Satu kelas Menu_Item merupakan sebahagian daripada satu kelas DrawingTools.
- Banyak atau beberapa kelas canvas merupakan sebahagian daripada satu kelas DrawingTools.
- Kelas Graphics merupakan perwarisan daripada kelas canvas.
- Satu kelas Menu_Item merupakan sebahagian daripada DrawingTools.
- Satu kelas Calculator, satu kelas moveController, satu kelas SizeAdjuster, satu kelas ColorAdjuster, satu kelas ArcAdjuster dan satu kelas PenAdjuster, semuanya merupakan sebahagian daripada (aggregation) satu kelas Menu_Item.

- Satu kelas moveButton merupakan hubungan (association) satu kepada satu bagi kelas moveController.
- Satu kelas mouse merupakan hubungan (association) satu kepada satu bagi kelas SizeAdjuster.
- Satu kelas color merupakan hubungan (association) satu kepada satu bagi kelas ColorAdjuster.
- Satu kelas protractor merupakan hubungan (association) satu kepada satu bagi kelas ArcAdjuster.
- Hubungan antara satu kelas calculator dengan satu kelas Graphic adalah calculate Perimeter/Area.
- Hubungan antara kelas move dengan kelas Graphic adalah move.
- Hubungan antara kelas Mouse dengan kelas Graphic adalah move.
- Hubungan antara kelas color dengan kelas Graphic adalah fill.
- Hubungan antara kelas protractor dengan kelas Graphic adalah move.
- Kelas Shape merupakan perwarisan daripada kelas Graphic.
- Satu kelas pen menghasilkan 0 hingga banyak kelas Drawing.
- Satu kelas Drawing menghasilkan 0 hingga banyak kelas Shape.

Bahagian III



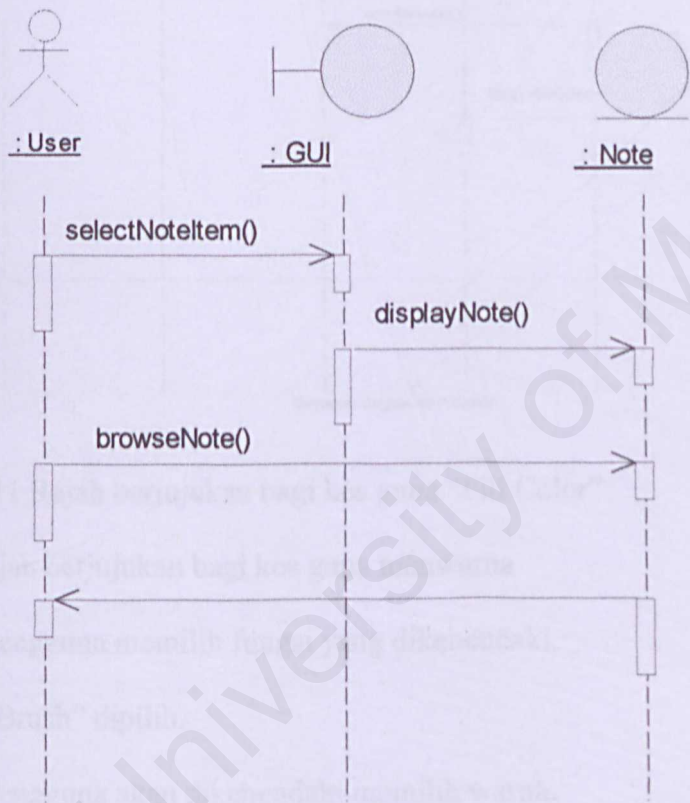
Rajah 5.9 Gambarajah kelas bahagian III

Dalam bahagian III, diagram menunjukkan:

- Banyak kelas Drawing mempunyai banyak kelas DrawingListener.
- Satu kelas Drawing “realize” satu kelas DrawingEvent.
- Satu kelas Shape terdiri daripada 0 hingga banyak kelas Point.
- Satu kelas Shape 2D dan satu kelas Shape 3D merupakan perwarisan daripada satu kelas Shape.
- Satu kelas circle, satu kelas square dan satu kelas triangle merupakan perwarisan daripada satu kelas Shape 2D.
- Satu kelas cube, satu kelas cuboid dan satu kelas pyramid merupakan perwarisan daripada satu kelas Shape 3D.

5.3 Rajah Berjukkan (Sequence Diagram)

Rajah ini memaparkan secara bergrafik bagaimana objek-objek berinteraksi satu sama lain melalui pemesejan di antara perlaksanaan bagi suatu kes guna atau operasi . Ia juga menunjukkan bagaimana mesej-mesej dihantar dan diterima di antara objek-objek berserta dengan susunan jujukan.



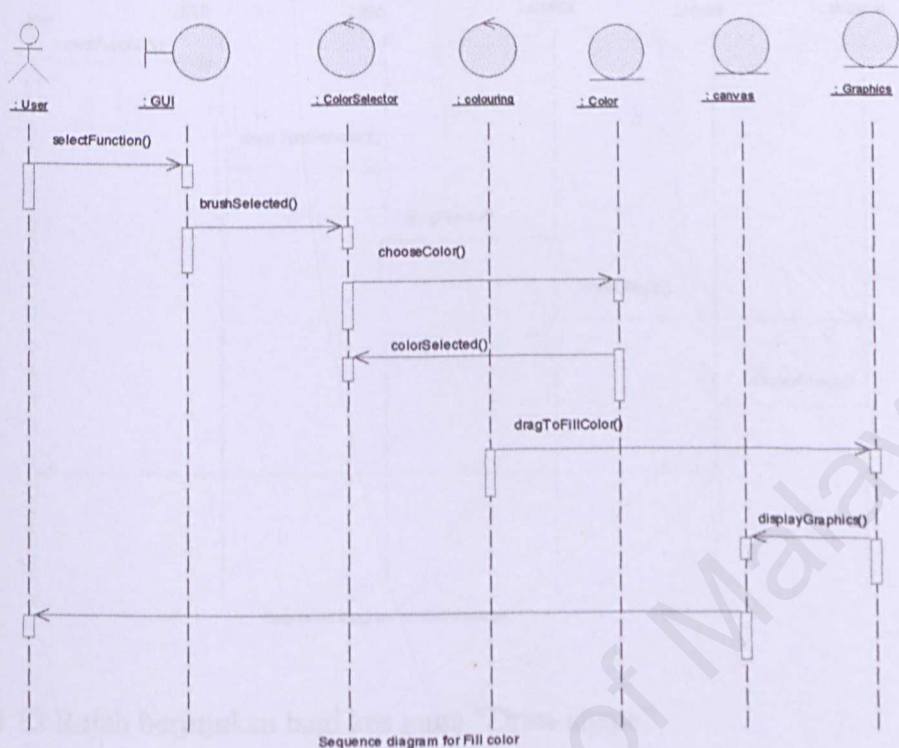
Sequence diagram for Browse notes

Rajah 5.10 Rajah berjujukan bagi kes guna“Browse notes”

Aliran rajah berjujukan bagi kes guna untuk Browsing Notes

- 1. Pengguna memilih Note dari Menu Utama.
- 2. Note dipaparkan.

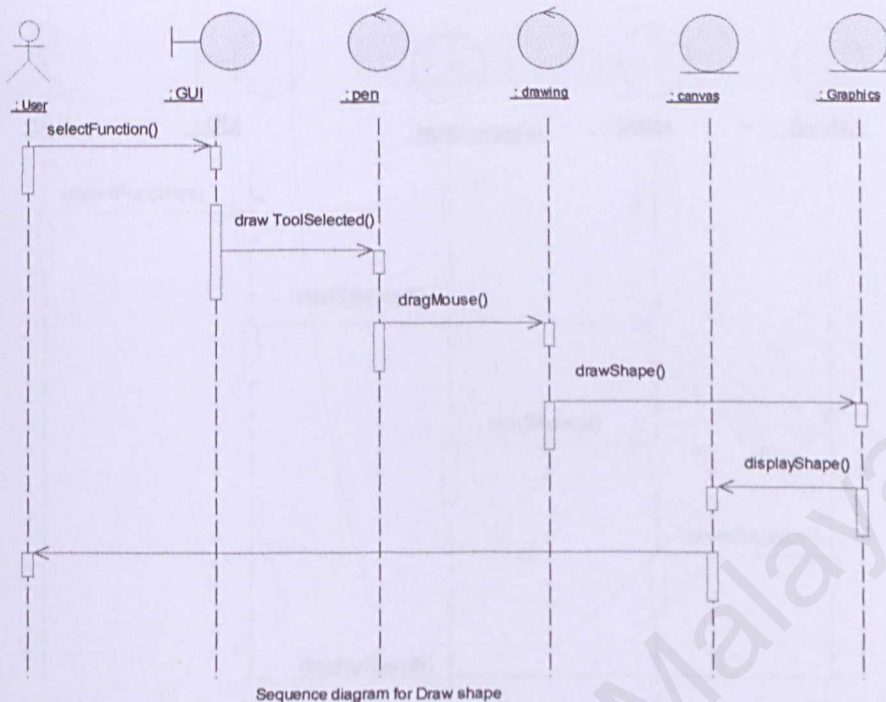
3. Pengguna membaca Note tersebut.”



Rajah 5.11 Rajah berjjukan bagi kes guna “Fill Color”

Aliran rajah berjjukan bagi kes guna menwarna

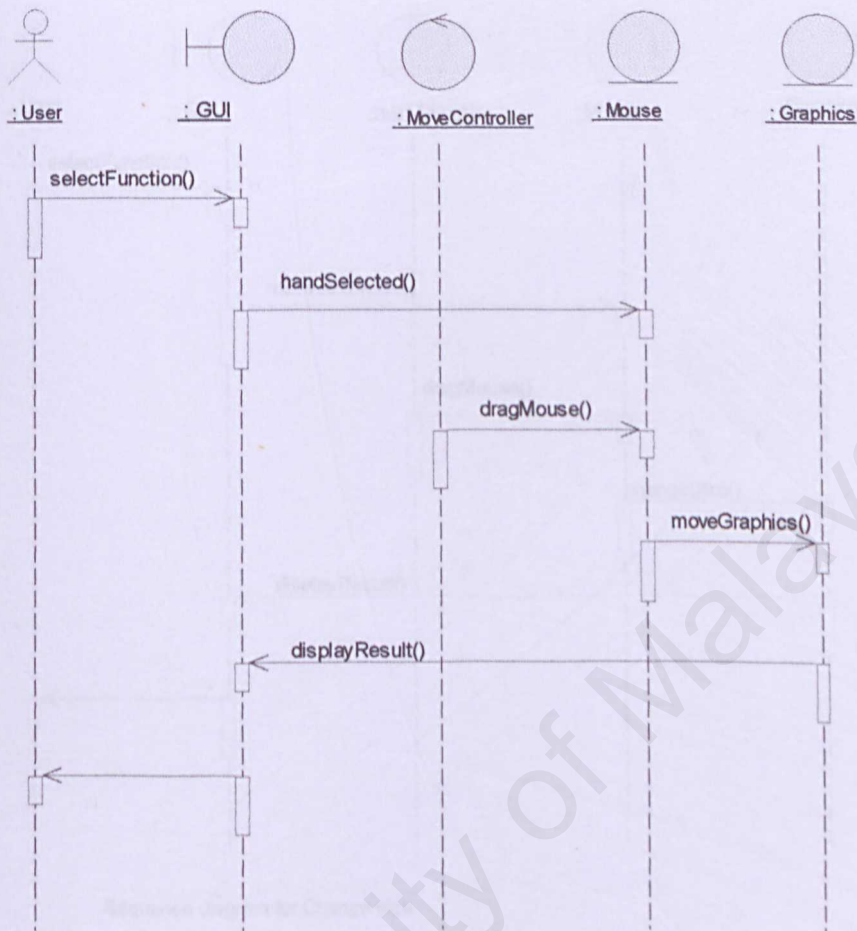
1. Pengguna memilih fungsi yang dikehendaki.
2. “Brush” dipilih.
3. Pengguna akan dikehendaki memilih warna.
4. Selepas warna dipilih, “drag” tetikus untuk mengisi warna bagi bentuk yang dilukis.
5. Grafik yang telah diisikan warna akan dipaparkan.



Rajah 5.12 Rajah berjjukan bagi kes guna “Draw shape”

Aliran rajah berjjukan bagi kes guna Drawing

1. Pengguna memilih fungsi yang dikehendaki.
2. Peralatan melukis dipilih.
3. “Drag” tetikus untuk melukis.
4. Melukis bentuk yang dikehendaki.
5. Bentuk yang dilukis dipaparkan.

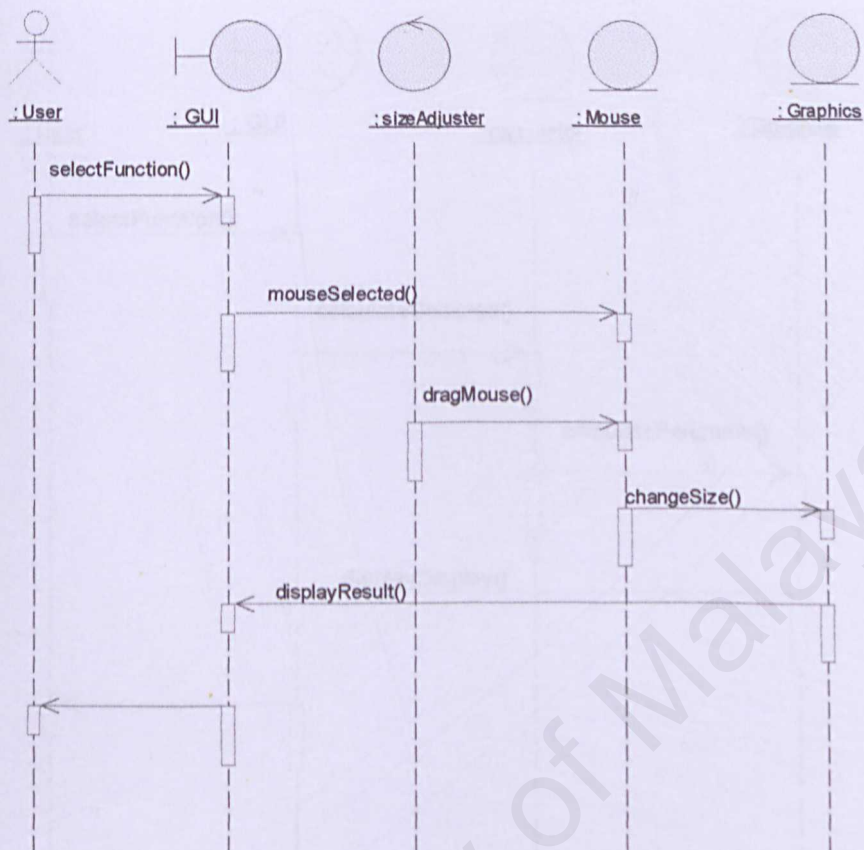


Sequence diagram for Move shape

Rajah 5.13 Rajah berjujukan bagi kes guna “Move shape”

Aliran rajah berjujukan bagi kes guna menggerakkan bentuk

1. Fungsi yang dikehendaki dipilih.
2. “hand” dipilih.
3. “Drag” tertikus terhadap bentuk yang dipilih.
4. Menggerakkan bentuk tersebut.
5. Memaparkan keputusan setelah bentuk digerakkan.

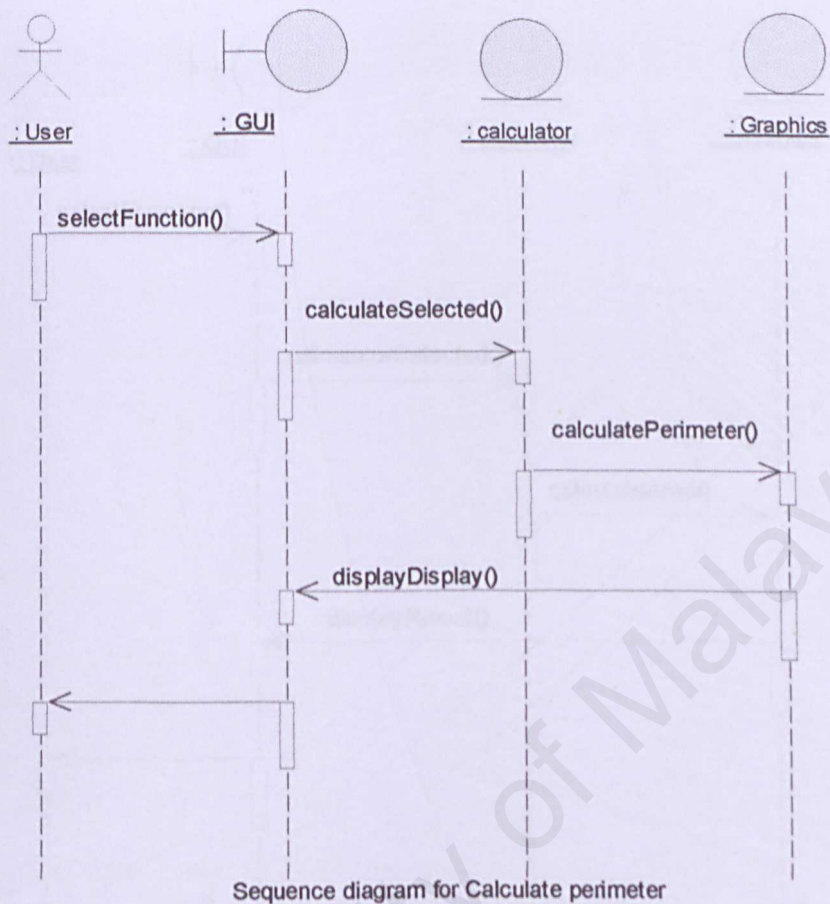


Sequence diagram for Change size

Rajah 5.14 Rajah berjujukan bagi kes guna “Change size”

Aliran rajah berjujukan bagi kes guna mengubah saiz bentuk

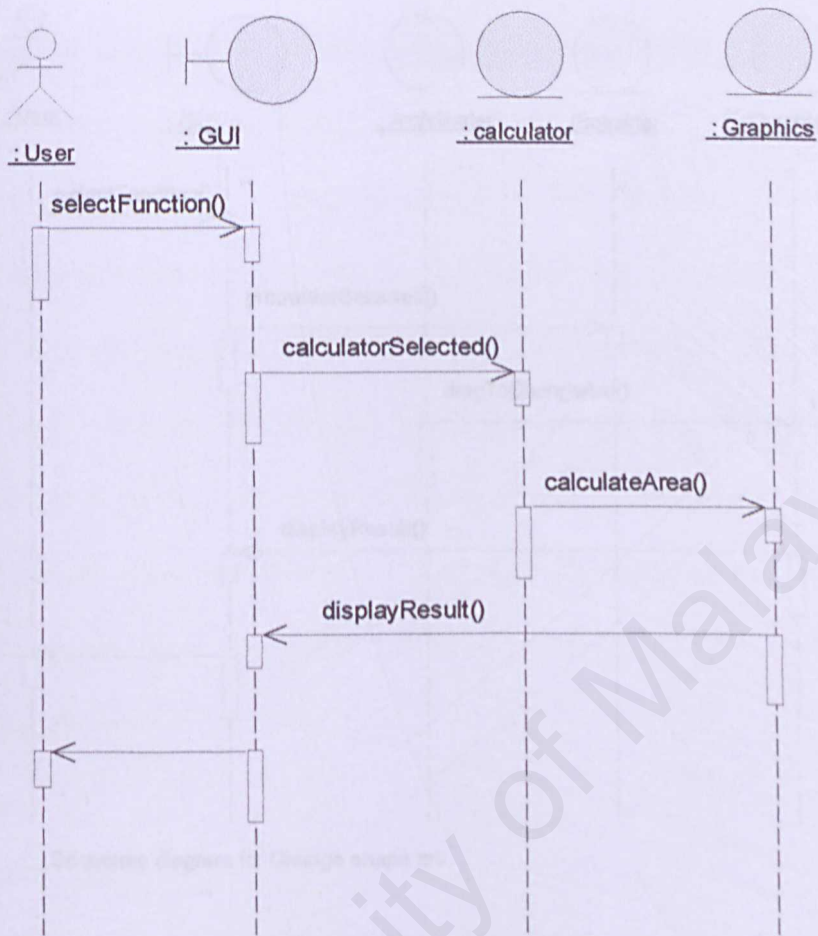
1. Pengguna memilih fungsi yang dikehendaki.
2. “Mouse” dipilih.
3. “Drag” tetikus terhadap bentuk yang dipilih.
4. Mengubah saiz bagi bentuk tersebut.
5. Memaparkan keputusan bagi bentuk yang saiznya telah berubah.



Rajah 5.15 Rajah berjujukan bagi kes guna “Calculate perimeter”

Aliran rajah berjujukan bagi kes guna mengira Perimeter

1. Pengguna memilih fungsi yang dikehendaki.
2. “Calculator” dipilih.
3. Perimeter bagi bentuk yang dipilih dikirakan perimeternya.
4. Keputusan perimeter dipaparkan.

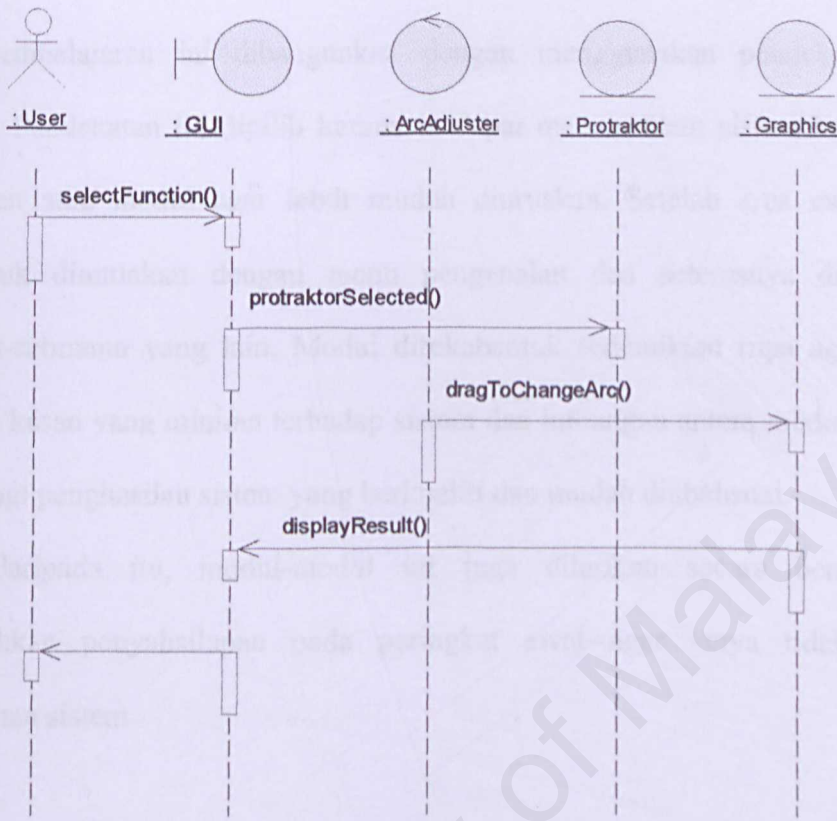


Sequence diagram for Calculate area

Rajah 5.16 Rajah berjjukan bagi kes guna “Calculate area”

Aliran rajah berjjukan bagi kes guna mengira luas bentuk:

1. Pengguna memilih fungsi yang dikehendaki.
2. “Calculator” dipilih.
3. Luas bagi bentuk yang dipilih dikirakan luasnya.
4. Keputusan perimeter dipaparkan.



Sequence diagram for Change shape arc

Rajah 5.17 Rajah berjujukan bagi kes guna “Change shape arc”

Aliran rajah berjujukan bagi kes guna mengubah sudut bentuk.

1. Pengguna memilih fungsi yang dikehendaki.
2. “Protraktor” yang digunakan untuk mengubah sudut dipilih.
3. “Drag” tetikus untuk mengubah sudut bagi bentuk yang telah dipilih.
4. Keputusan bagi bentuk yang sudut yang telah diubah dipaparkan.

5.4 Rekabentuk Struktur Program

Pakej pembelajaran ini dibangunkan dengan menggunakan pendekatan “Atas ke Bawah”. Pendekatan ini dipilih kerana ia dapat memecahkan sistem kepada beberapa komponen atau modul agar lebih mudah diuruskan. Setelah aras menu ditentukan rekabentuk dimulakan dengan menu pengenalan dan seterusnya disusuli dengan submenu-submenu yang lain. Modul direkabentuk sedemikian rupa agar ianya dapat memberi kesan yang minima terhadap sistem dan hubungan antara modul-modul adalah terhad bagi penghasilan sistem yang berkualiti dan mudah diubahsuai.

Selain daripada itu, modul-modul ini juga dilarikan secara berasingan untuk memudahkan penyahsilapan pada peringkat awal agar ianya tidak menjejaskan keseluruhan sistem.

5.5 Rekabentuk Skrin

Rekabentuk skrin adalah satu isu yang harus dititikberatkan. Bagi sistem bergrafik, ianya penting agar ia kelihatan menarik dan mudah difahami oleh pengguna. Rekabentuk sesuatu sistem mestilah mesra-pengguna (“user-friendly”). Pada keseluruhannya sistem ini boleh dikatakan mempunyai latarbelakang yang menarik dan berwarna-warni.

Bagi pilihan menu, kaedah butang tekan yang konsisten digunakan agar memudahkan pengguna. Secara keseluruhannya, pengguna menggunakan tetikus untuk berinteraksi dengan pakej pembelajaran ini.

5.6 Rekabentuk Sistem

Rekabentuk sistem ini dibangunkan adalah untuk menarik minat dan perhatian pelajar-pelajar berumur 10-12 terhadap pakej pembelajaran ini. Dengan itu, rekabentuk sistem ini boleh dibahagikan kepada bahagian iaitu

- i) Struktur sistem
- ii) Rekabentuk antaramuka sistem

5.6.1 Struktur Sistem

Struktur sistem adalah alat yang digunakan dalam merekabentuk sistem ini. Ianya merupakan interaksi dan hubungan yang berlaku antara modul-modul yang bergantung di antara satu sama lain.

5.6.2 Rekabentuk Antaramuka Sistem

Objektif utama rekabentuk antaramuka sistem adalah untuk menyediakan cara terbaik bagi pengguna berinteraksi dengan sistem. Antaramuka yang baik akan meningkatkan pemahaman pengguna dan kualiti persembahan kerja.

Rekabentuk antaramuka yang baik adalah rekabentuk antaramuka yang dapat memenuhi objektif dengan cekap, tepat segala fungsi-fungsinya, konsisten, mudah dan menarik. Ianya juga perlu memahami bagaimana tindak balas pengguna terhadap unsur-unsur yang berbeza yang disediakan atau diwujudkan pada antaramuka tersebut.

Rekabentuk Antaramuka Sistem merupakan salah satu langkah yang terkandung di dalam rekabentuk sistem. Ia bertujuan untuk menyediakan kaedah yang terbaik bagi interaksi antara pengguna dan komputer atau dikenali sebagai Interaksi Komputer

Manusia (*Human Computer Interaction*).Rekabentuk antaramuka yang baik merupakan satu elemen yang dititikberatkan di dalam pembangunan sistem kerana ia memberi kesan yang besar kepada kejayaan sesuatu sistem serta kepuasan pengguna terhadap sistem tersebut. Pengguna tidak mahu meluangkan masa yang banyak untuk mempelajari sesebuah sistem sebaliknya mereka lebih cenderung untuk menjadikan komputer sebagai peralatan yang memudahkan kerja mereka .Oleh itu, antaramuka baik boleh membantu dalam mencapai sasaran ini.

Antaramuka pengguna boleh dibina dalam bentuk yang berasaskan borang (*Form – based*) atau teks(*Textual*) . Akan tetapi , kebanyakan komputer pada masa kini mengandungi pemproses yang berkuasa dan paparan warna yang beresolusi tinggi. Maka antaramuka pengguna bergrafik(GUI) adalah amat sesuai digunakan di mana ia menyokong skrin berwarna yang beresolusi tinggi dan berinteraksi dengan menggunakan tetikus serta papan kekunci . Oleh sebab itu, rekabentuk antaramuka sistem ini akan berasaskan pendekatan GUI untuk mencapai interaksi yang optimum .

Jadual menggariskan beberapa ciri-ciri asas pada GUI.

Ciri-Ciri	Huraian
Tetingkap	Tetingkap berganda yang membolehkan maklumat yang berbeza dipaparkan serentak pada skrin pengguna
Ikon	Ikon mewakili jenis maklumat yang berbeza . Ikon-ikon ini mewakili fail atau

	proses tertentu
Menu	Arahan dipilih dari menu
Penunjuk	Peralatan penunjuk seperti tetikus digunakan untuk memilih pilihan dari menu atau menanda item yang dimuatkan di dalam tetingkap
Grafik	Grafik merupakan imej atau gambar yang ditambah bersama terus ke dalam sebuah skrin, ikon dan sebagainya.

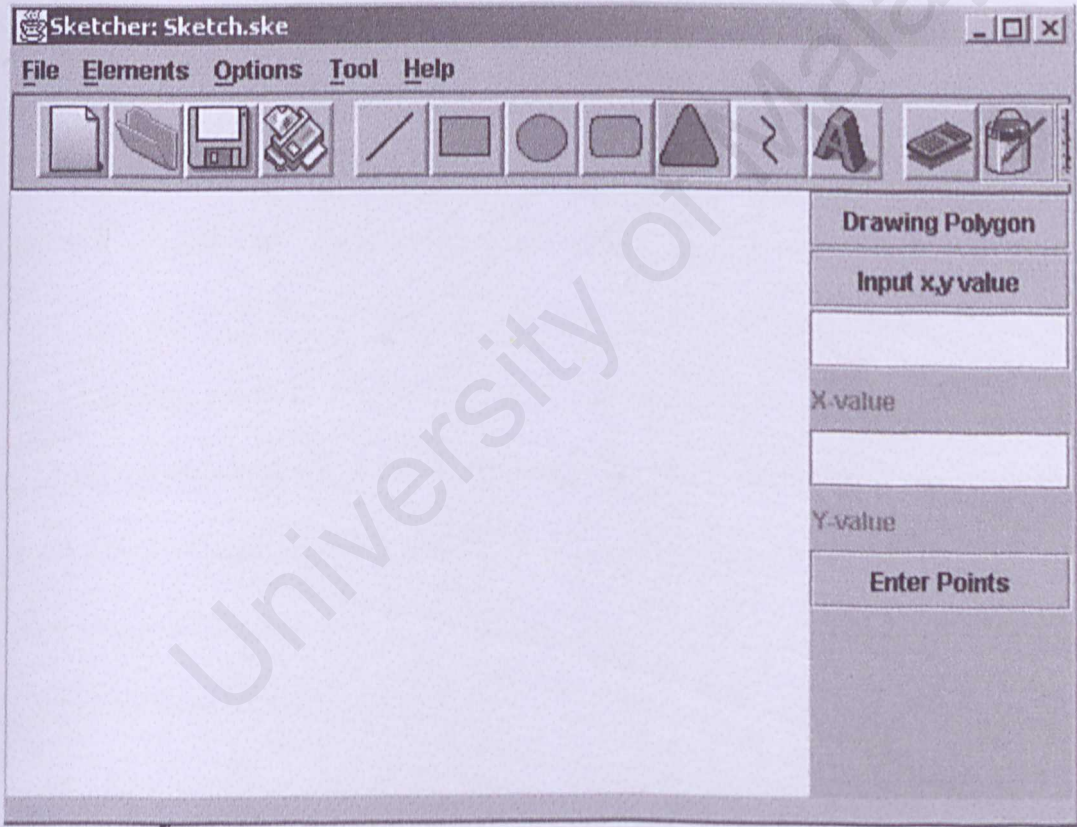
Jadual 5.1: Ciri-ciri asas pada GUI

Di samping itu, GUI juga mempertimbangkan keperluan, pengetahuan dan keupayaan penggunaan sistem. Jadual menyatakan beberapa prinsip utama yang boleh diseduaikan untuk semua GUI pada sistem.

Prinsip	Huraian
Kemesraan pengguna	Antaramuka harus menggunakan syarat dan konsep yang berasaskan pengguna
Konsisten	Antaramuka haruslah konsistensi di mana operasi dipersembahkan dengan cara yang sama
Minimum kejutan	Pengguna haruslah tidak terkejut dengan kelakuan sistem
Kebolehpulihan	Antaramuka harus memasukkan cara –cara

	yang betul bagi membenarkan pengguna memulihkan ralat mereka
Panduan pengguna	Antaramuka haruslah dibekalkan bersama borang-borang yang menyediakan panduan kepada pengguna

Jadual 5.2: Prinsip-prinsip utama antaramuka pengguna



Rajah 5.18: Rekabentuk antaramuka bagi “DrawingTool” sistem ini

5.7 Ringkasan Bab 5

Bab ini menerangkan tentang fasa rekabentuk yang meliputi rekabentuk struktur program, skrin dan sistem. Bab ini juga merangkumi rajah kelas dan rajah berjujukan sebagai penerangan kepada sistem yang akan dibangunkan. Rekabentuk antaramuka pengguna banyak mempertimbangkan “mesra-pengguna” memandangkan pengguna sasaran adalah pelajar-pelajar sekolah rendah.

Bab 6 IMPLEMENTASI DAN PEMBANGUNAN SISTEM

6.1 Pengenalan

Fasa ini melibatkan semua modul-modul yang telah direkabentuk diintegrasikan kepada penghasilan sebuah sistem berasaskan keperluan-keperluan yang disenaraikan atau ia boleh didefinisikan sebagai penterjemahan perwakilan yang dibuat dalam fasa rekabentuk kepada produk sebenar. Dalam melaksanakan fasa ini, masa yang banyak telah digunakan. Proses pada fasa ini adalah rumit dan memerlukan perhatian yang banyak dan teliti.

6.2 Persekitaran Pembangunan

Dalam persekitaran pembangunan, terdapat dua perkara utama yang perlu dipertimbangkan iaitu: perkakasan dan perisian.

6.2.1 Konfigurasi Perkakasan

Perkakasan yang digunakan untuk pembangunan sistem ini adalah:

- Pemprosesan Pentium III 550 MHz
- 64MB SDRAM
- 10 GB Hard Disk
- 1.44 MB Floppy Disk
- 40x CDROM
- 15' monitor
- Komponen-komponen PC lain yang sempadan

6.2.2 Konfigurasi Perisian

Konfigurasi perisian yang digunakan untuk membangunkan sistem ini adalah seperti disenaraikan di bawah:

Perisian	Penjelasan
Windows 2000 Professional	Sebagai sistem pengoperasian.
JDK 1.3.1	Kit pembangunan persekitaran bahasa pengaturcaraan JAVA
JCreator LE 2.0	Sebagai penyunting , pengkompil dan pelaksanaan kod-kod program JAVA.
Notepad	Sebagai penyunting kod-kod program.
MS Command Prompt	Untuk mengkompil dan melarikan kod-kod program JAVA.

Jadual 6.1 Jenis-jenis perisian yang digunakan dalam pembangunan perisian

Sebelum proses pengekodan , pemasangan (install) JDK 1.3.1 (Java Development Kit version 1.3.1) harus dilakukan. JDK diperlukan untuk menghasilkan perisian Java kerana JDK mengandungi pengkompil, mesin maya (virtual mesin), librari kelas (class library), dokumentasi dan contoh-contoh.

Kemudian, pemasangan JCreator LE 2.0 sebagai penyunting kod dilakukan dan perisian ini boleh didapati secara percuma sebagai ‘freeware’ dengan memuatturunnya(download) dari internet (www.download.com)

6.3 Pembangunan Program

Pembangunan program merupakan proses menghasilkan program yang diperlukan untuk memenuhi keperluan-keperluan fungsi yang dicadangkan. Pembangunan program terdiri daripada lima langkah yang perlu diikuti:

1. Membuat tinjau semula ke atas dokumentasi program
2. Merekabentuk program
3. Pengekoden program
4. Menjalankan pengujian program
5. Melengkapkan dokumentasi program

6.3.1 Membuat tinjau semula ke atas dokumentasi program

Langkah pertama dalam pembangunan program adalah menilai semula kod aturcara yang sedia ada kerana hendak mengubah atau menggantikannya dengan yang baru atau menggunakannya semula di dalam aplikasi yang lain. Misalnya dokumentasi program bagi sistem ini terdiri daripada keterangan ringkas mengenai sesuatu proses dan dokumen maklumat-maklumat yang terkumpul. Dokumen-dokumen ini dapat membantu dalam memahami terhadap sistem yang perlu dibangunkan dengan lebih mendalam semasa pengekodan.

6.3.2 Merekabentuk program

Rekabentuk program harus dilaksanakan selepas membuat tinjau terhadap dokumen-dokumen maklumat dan program di mana ini merupakan langkah yang kedua dalam pembangunan sistem. Semasa langkah ini dijalankan, keputusan bagaimana program

dapat disempurnakan harus ditetapkan dan apa yang harus dibuat adalah membangunkan satu penyelesaian yang logical terhadap masalah pengaturcaraan.

6.3.3 Pengekoden program

Pengekoden program merupakan proses menulis arahan-arahan program yang akan mengimplementasikan rekabentuk program. Spesifikasi rekabentuk mesti dapat ditranslasikan ke dalam format yang dapat dibaca oleh mesin. Kod-kod program mempunyai fungsi yang sedemikian. Jika rekabentuk program dipersembahkan dengan dengan baik , maka mesin akan dapat menyempurnakan tugas tersebut dan menghasilkan output yang diinginkan.

6.3.4 Pengujian program

Semasa peringkat pengujian program, program akan diuji secara menyeluruh untuk memastikan ia dapat berfungsi dengan baik. Pelbagai jenis pengujian dapat dijalankan untuk menguji program secara individu atau secara bergabungan.

6.3.5 Dokumentasi program

Dokumentasi program yang sesuai dan lengkap adalah penting untuk kejayaan sesuatu sistem beroperasi dan juga dapat menyenangkan penyelenggaraan. Selain dokumentasi program, manual pengguna juga harus disertakan kerana ia amat diperlukan oleh pengguna akhir supaya mereka lebih memahami bagaimana sistem itu dapat digunakan.

6.4 Pengekoden program

6.4.1 Pendekatan pengkodean

Pendekatan pengkodean yang digunakan adalah kaedah atas-bawah. Berikut adalah contoh-contoh kod sebahagian yang dihasilkan berdasarkan pendekatan pengkodean atas bawah secara berperingkat-peringkat bagi membangunkan perisian ini.

Langkah pertama: Penghasilan tettingkap (windows) asas

```
public void init()
{
    window = new SketchFrame("Sketcher", this); // Create the app window
    Toolkit theKit = window.getToolkit(); // Get the window toolkit
    Dimension wndSize = theKit.getScreenSize(); // Get screen size

    // Set the position to screen center & size to 2/3 screen size
    window.setBounds(wndSize.width/6, wndSize.height/6, // Position
        2*wndSize.width/3, 2*wndSize.height/3); // Size
}
```

Langkah kedua: Menambahkan komponen-komponen berfungsi seperti butang, menu, bar peralatan (tool bar) ke dalam tettingkap utama

Berikut merupakan contoh kod dalam operasi menambah komponen-komponen ke dalam tettingkap utama:

i) Menambahkan menu pada frame utama

```
JMenu fileMenu = new JMenu("File"); // Create File menu
JMenu elementMenu = new JMenu("Elements"); // Create Elements menu
```

```
JMenu optionsMenu = new JMenu("Options"); // Create options menu
```

```
JMenu helpMenu = new JMenu("Help"); // Create Help menu
```

```
JMenu toolMenu = new JMenu("Tool"); // Create Tool menu
```

ii) Menambahkan butang pada frame utama

```
JButton drawBtn = new JButton();
```

```
JButton enterBtn = new JButton();
```

```
JButton resetBtn = new JButton();
```

iii) Menambahkan 'toolbar' di dalam frame utama

```
private JToolBar toolBar = new JToolBar(); // Window toolbar
```

```
getContentPane().add(toolBar, BorderLayout.NORTH);
```

Langkah ketiga: Mengimplementasikan 'Event Listener' pada paras program yang lebih rendah

'Event Listener' merupakan sintaks dalam kod JAVA yang memberi fungsi sama ada kepada butang, komen dan sebagainya.

i) Contoh aksi bagi komen di menu

```
// Handle About menu action events
```

```
public void actionPerformed(ActionEvent e)
```

```
{
```

```
if(e.getSource() == aboutItem)
```

```
{
```

```
// Create about dialog with the menu item as parent
```



```

JOptionPane.showMessageDialog(this, // Parent
    "Sketcher Copyright 2002", // Message
    "About Sketcher", // Title
    JOptionPane.INFORMATION_MESSAGE); // Message type
}

```

ii)Persembahan aksi bagi butang di kalkulator

```

public void actionPerformed( ActionEvent e )
{
    //store the instance of the button that was pressed
    ButtonComponent button = (ButtonComponent) e.getSource();

    //get the button's label
    String s = button.getLabel().trim();

    //determine if the button was a number or operator button
    if ( Character.isDigit( s.charAt(0) ) )
        handleNumber( s );
    else
        calculate( s );
}

```

Langkah keempat: Menghasilkan kelas bagi objek yang hendak dilukis

i) Menghasilkan objek segiempat

// Nested class defining a rectangle

```
public static class Rectangle extends Element
```

```
{  
  
    public Rectangle(Point start, Point end, Color color)  
  
    {  
  
        super(color);
```

```
position = new Point(Math.min(start.x, end.x), Math.min(start.y, end.y));
```

```
rectangle = new Rectangle2D.Double(  
    origin.x,  
    origin.y,  
    Math.abs(start.x - end.x), // Width  
    Math.abs(start.y - end.y)); // & height  
}
```

```
public java.awt.Rectangle getBounds()  
  
    {  
  
        return getBounds(rectangle.getBounds());  
  
    }
```

```
public void draw(Graphics2D g2D)  
  
    {  
  
        draw(g2D, rectangle); // Call the base draw method  
  
    }
```

```
private void writeObject(ObjectOutputStream out) throws IOException
```

```
{
```



```

        out.writeDouble(rectangle.width);

        out.writeDouble(rectangle.height);

    }

    private void readObject(ObjectInputStream in)

        throws IOException, ClassNotFoundException

    {

        double width = in.readDouble();

        double height = in.readDouble();

        rectangle = new Rectangle2D.Double(0,0,width,height);

    }

    private Rectangle2D.Double rectangle;

}

```

ii) Menghasilkan objek bulatan

// Nested class defining a circle

public static class Circle extends Element

```

{

    public Circle(Point center, Point circum, Color color)

    {

        super(color);

        statusBar.setText( center.toString() + ":" + circum.toString() );

        //System.out.println("

        // Radius is distance from center to circumference

```

```

radius = center.distance(circum);

position = new Point(center.x - (int)radius, center.y - (int)radius);

circle = new Ellipse2D.Double(origin.x, origin.y,    // Position - top-left
                               2.*radius, 2.*radius ); // Width & height
}

public double getRadius()
{return radius;}

private double radius;

//Calculate perimeter of circle
public double perimeter()
{
return Math.PI*radius*2;
}

//Calculate area of circle
public double area()
{ return Math.PI*radius*radius;
}

public java.awt.Rectangle getBounds()
{
return getBounds(circle.getBounds());
}

```



```

    }

    public void modify(Point center, Point circum)
    {
        double radius = center.distance(circum);

        position.x = center.x - (int)radius;
        position.y = center.y - (int)radius;

        circle.width = circle.height = 2*radius;

        this.radius = radius; System.out.println(radius);
    }

    public void draw(Graphics2D g2D)
    {
        //g2D.setStroke(new BasicStroke(2.0f));

        draw(g2D, circle); // Call the base draw method
    }

    private void writeObject(ObjectOutputStream out) throws IOException
    {
        out.writeDouble(circle.width);
    }

    private void readObject(ObjectInputStream in)
        throws IOException, ClassNotFoundException
    {
        double width = in.readDouble();

        circle = new Ellipse2D.Double(0,0,width,width);
    }

```

```

}

private Ellipse2D.Double circle;

}

```

Langkah kelima: Pengekodan bagi interaksi pengguna menggunakan tetikus dengan objek yang hendak dilukis iaitu mengimplementasikan 'Event Listener' pada paras program yang lebih tinggi

i) Memberikan aksi apabila tetikus di'drag' kan

```

public void mouseDragged(MouseEvent e)

{ //statusPane.setText("(" + e.getStart() + ", " + e.getLast() + ")");

    last = e.getPoint(); // Save cursor position

    int modifier = e.getModifiers(); // Get modifiers

    if((modifier & e.BUTTON1_MASK) != 0 &&

        (theApp.getWindow().getElementType() != TEXT) && (mode == NORMAL))

    {

        if(tempElement == null) // Is there an element?

            tempElement = createElement(start, last); // No, so create one

        else

        {

            tempElement.draw(g2D); // Yes - draw to erase it

            tempElement.modify(start, last); // Now modify it

        }

        tempElement.setStroke( theApp.getWindow().currentWidth );
    }

```



```

tempElement.setFilled(theApp.getWindow().isFilled);
tempElement.setColor( theApp.getWindow().getElementColor());
tempElement.draw(g2D);          // and draw it
}

else if((modifier & e.BUTTON1_MASK) != 0 &&
(theApp.getWindow().getElementType() == ERASER) && (mode == NORMAL))
{
    if(tempElement == null)          // Is there an element?
        tempElement = createElement(start, last);    // No, so create one
    else
    {
        g2D.setColor(Color.white);
        tempElement.draw(g2D);      // Yes - draw to erase it
        tempElement.modify(start, last);    // Now modify it
    }
    tempElement.draw(g2D);          // and draw it
}
}

```

ii) Menghasilkan aksi apabila tetikus dilepaskan

```

public void mouseReleased(MouseEvent e)
{
    int modifier = e.getModifiers();    // Get modifiers
    if(e.isPopupTrigger())
    {

```

```

start = e.getPoint();

if(highlightElement == null)

    theApp.getWindow().getPopup().show((Component)e.getSource(),      start.x,
start.y);

else

    elementPopup.show((Component)e.getSource(), start.x, start.y);

    start = null;
}

else if((modifier & e.BUTTON1_MASK) != 0 &&
(theApp.getWindow().getElementType() != TEXT) && mode == NORMAL)
{
    if(tempElement != null)
    {
        theApp.getModel().add(tempElement); // Add element to the model
        tempElement = null;
    }
}

else if(mode == MOVE || mode == ROTATE)
{
    if(selectedElement != null)

        repaint();

        mode = NORMAL;
}

if(g2D != null)

```



```

{
    g2D.dispose();           // Release graphic context resource

    g2D = null;              // Set it to null
}

start = last = null;       // Remove the points

selectedElement = tempElement = null; // Reset elements
}

```

Langkah keenam: Memberikan aksi kepada objek yang telah dilukis seperti pergerakan

(move)

```

selectedElement.draw(g2D);           // Draw to erase the element

selectedElement.move(last.x-start.x, last.y-start.y); // Move it

selectedElement.draw(g2D);           // Draw in its new position

start = last;                        // Make start current point

```

6.4.2 Corak pengekodan

Pengaturcaraan banyak melibatkan kreativiti. Namun, tidak kira apa bahasa pengaturcaraan yang digunakan, setiap komponen program melibatkan 3 aspek utama:

Struktur kawalan

Kebanyakan struktur kawalan untuk komponen dicadangkan dalam senibina dan rekabentuk, maka diterjemahkan kepada kod. Bagi sebarang jenis rekabentuk, adalah penting bagi struktur program menunjukkan rekabentuk struktur kawalan supaya komponen mudah dari atas ke bawah.

Contohnya, dalam kod program bagi sistem ini, struktur kawalan digunakan ditunjukkan di bawah:

```
if(source == moveItem)

{
    mode = MOVE;

    selectedElement = highlightElement;
}

else if(source == deleteItem)

{
    if(highlightElement != null)           // If there's an element
    {
        theApp.getModel().remove(highlightElement); // then remove it
        highlightElement = null;           // Remove the reference
    }
}

else if(source == rotateItem)

{
    mode = ROTATE;

    selectedElement = highlightElement;
}

else if(source == sendToBackItem)

{
    if(highlightElement != null)
    {
```



```

theApp.getModel().remove(highlightElement);

theApp.getModel().add(highlightElement);

highlightElement.setHighlighted(false);

highlightElement = null;

repaint();

}

```

Algoritma

Rekabentuk program biasanya spesifikasikan algoritma-algoritma yang akan digunakan di dalam pengkodan. Algoritma yang dijalankan dalam pembangunan perisian ini adalah seperti dinyatakan di bawah:

- Merekabentuk rekabentuk antaramuka bergrafik
- Menyusunkan komponen-komponen GUI seperti bebutang, menu dan frame paparan supaya mengikut kekonsistenan rekabentuk antaramuka.
- Selepas itu , setiap komponen yang mempunyai fungsi yang unit diberikan aksinya masing-masing. Contohnya melalui panggilan fungsi 'actionPerformed' apabila 'addActionListener' ditambah ke dalam satu komen.
- Akhirnya, menambah *main()* metod ke dalam aplikasi tersebut..
- Setelah selesai keseluruhan pengkodan tersebut, setiap fail tersebut disimpan sebaga '*.java' dan diletakkan bersama dalam satu folder yang sama.

Struktur data

Dalam menulis program, data harus disimpan dan diformatkan supaya pengurusan data dan juga manipulasi data adalah mudah. Contoh teknik struktur data yang digunakan dalam kod-kod program sistem ini adalah penggunaan array seperti yang ditunjukkan di bawah:

```
public Polygon(int xPlot[], int yPlot[], Color color)
```

6.4.3 Dokumentasi kod

Dokumentasi kod dimulakan dengan memilih pembolehubah yang sesuai , kemudian dihubungkan antara satu sama lain dan akhirnya diorganisasikan kepada program. Gunakan baris yang kosong atau indentasi (indentation) supaya komen yang ditambah boleh dibezakan daripada kod-kod program.

➤ Dokumentasi dalaman

Komen-komen dalam pengaturcaraan memberi satu panduan yang jelas apabila fasa penyelenggaraan. Komen memudah komunikasi antara pembangun dan kod-kod sumber. Tujuan penghasilan sesuatu fungsi digalakkan diselit antara kod-kod sumber jika perlu untuk menerangkan fungsi tersebut supaya memudahkan penyelenggaraan pada masa depan.

➤ Naming Convention

Naming Convention membekalkan identifikasi yang senang kepada pengaturcara.

➤ Modulariti

Untuk mengurangkan kekompleksan sesuatu proses pembangunan, subsistem atau sistem dipecahkan kepada modul-modul yang lebih kecil untuk menyenangkan pengimplementasian dan mengurangkan kekompleksan.

6.5 Ringkasan Bab 6

Bab ini menjelaskan tentang proses pembangunan dan pengekodan yang telah dijalankan. Ini melibatkan penterjemahan perwakilan yang dibuat dalam fasa rekabentuk kepada produk yang sebenar. Proses ini merupakan proses yang paling lama dan mengambil masa. Perisian utama yang digunakan untuk membangunkan pakej ini adalah pengedit program JCreator LE 2.0 dan bahasa pengaturcaraan JAVA. Pertimbangan yang telah dikenalpasti dan diikuti dalam pengekodan adalah kaedah pengekodan atas-bawah untuk menghasilkan kod yang mudah dibaca, diganti, tidak terlalu kompleks dan mesti dipiawaikan.

Bab 7 PENGUJIAN

7.1 Pengenalan

Dalam melaksanakan pengujian terhadap sistem, pengguna menguji keberkesanan sesuatu aturcara itu menjalankan fungsi seperti yang telah dikenalpasti. Tujuan pengujian ini dijalankan adalah untuk mengesan kesilapan. Maka matlamat pengujian akan tercapai hanya apabila pembangun menemui kesilapan atau kegagalan. Memang tidak boleh dinafikan bahawa dalam setiap pembangunan, mesti terdapat kesilapan yang tidak dapat dicari dan cara penyelesaian yang berkesan haruslah dicari untuk membetulkan kesilapan ini. Fasa pengujian merupakan perkara yang paling penting untuk memastikan sistem itu telah mencapai objektifnya dan memenuhi keperluan pengguna. Sistem yang berkualiti dan berkesan dikatakan mampu dijalani oleh apa jua jenis pengujian. Di bawah disenaraikan objektif-objektif dalam pengujian secara umum:

- Mengenalpasti ralat- Pemeriksaan terperinci dilakukan ke atas setiap fungsi dengan mencuba dengan pelbagai kemungkinan kesilapan yang mungkin wujud. Sebagai contohnya, meletakkan nilai pembolehubah yang palsu dalam fungsi.
- Memperbaiki ralat- Ralat yang dikenalpasti boleh diperbaiki dengan mengatucara semula fungsi yang mengandungi kesilapan dan disemak semula. Sebab-sebab berlakunya ralat dikenalpasti dan dibetulkan.
- Ujian regresi – Ujian ini dilakukan memastikan bahawa pembaikan ralat telah dilakukan sepenuhnya dan tidak akan mengganggu fungsi lain dalam sistem.

7.2 Jenis-jenis Ralat

7.2.1 Ralat algoritma

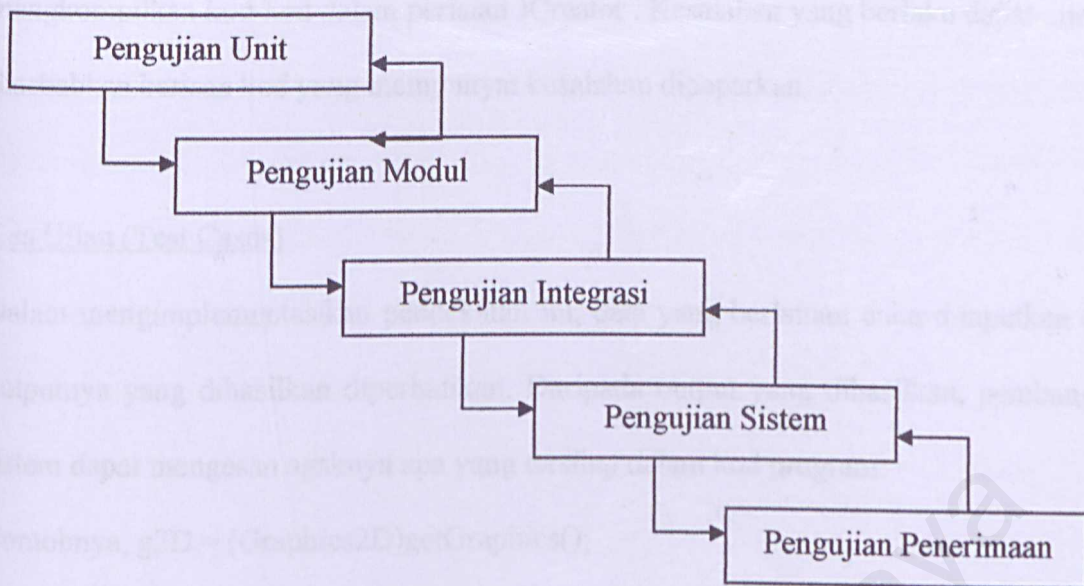
Ralat algoritma merupakan ralat yang berlaku apabila komponen atau logic tidak menghasilkan output yang sebenar bagi input yang diberi. Contoh ralat algoritma semasa pembangunan pakej ini adalah semasa polygon diharapkan sebagai outputnya, tetapi apa yang muncul di skrin adalah bulatan. Ini mungkin disebabkan oleh kesilapan semasa langkah pemprosesan.

7.2.2 Ralat pengiraan dan ketepatan

Ralat ini berlaku apabila implementasi formula adalah salah ataupun tidak tepat. Contohnya ralat pengiraan yang berlaku semasa pembangunan sistem ini adalah pengiraan perimeter dan luas bulatan yang salah menyebabkan output yang dihasilkan juga salah.

Pengujian dilakukan peringkat demi peringkat untuk memastikan unit-unit yang dibangunkan dapat dilarikan dengan sempurna secara berperingkat-peringkat. Pengujian terhadap modul dilaksanakan setelah sistem selesai dibangunkan oleh pembangun sendiri. Untuk pengujian keseluruhan pakej, pengujian dilakukan oleh penyelia, rakan-rakan dan bakal pengguna sistem. Penyelenggaraan semula sistem dilakukan setelah fasa pengujian selesai.

Carta: Peringkat-peringkat pengujian



Gambarajah 7.1 : Peringkat-peringkat pengujian

7.3 Pengujian Unit

Pengujian unit merujuk kepada pengujian kefungsi unit yang terdapat dalam sistem. Contohnya dalam pakej ini, butang bulatan dipilih dan diuji samada apabila tetikus 'drag' pada tertingap akan menghasilkan bulatan atau tidak. Ini kerana setiap fungsi unit dibangunkan dan diprogramkan secara berasingan dan ia dibiarkan dilarikan secara berasingan. Setiap kod dan komponen program diuji dan dipastikan betul dalam pengujian unit. Dalam pengujian unit, bebutang yang terdapat dalam setiap skrin diuji dan dipastikan berfungsi sebagaimana yang diinginkan.

Dua strategi yang digunakan untuk melaksanakan pengujian unit adalah:

Meninjau semula kod program yang ditulis

Dengan menggunakan strategi ini, setiap baris kod disemak semula untuk mengesan sebarang ralat semantik dan sintak yang terdapat dalam program dengan

mengkompilkan kod-kod dalam perisian JCreator . Kesalahan yang berlaku dapat dijejak disebabkan barisan kod yang mempunyai kesalahan dipaparkan.

Kes Ujian (Test Cases)

Dalam mengimplementasikan pendekatan ini, data yang berlainan cuba diinputkan dan outputnya yang dihasilkan diperhatikan. Daripada output yang dihasilkan, pembangun sistem dapat mengesan agaknya apa yang tersilap dalam kod program.

Contohnya, `g2D = (Graphics2D) getGraphics();`

`//g2D.drawString("adfd",10,10);` // Get graphics context

`g2D.setXORMode(getBackground());` // Set XOR mode

`g2D.setPaint(theApp.getWindow().getElementColor());` // Set color

`//g2D.drawString("adfd",10,10);` merupakan satu kes ujian.

7.4 Pengujian Modul

Dalam ujian modul, setiap fungsi unit yang telah diuji secara bersendirian digabungkan antara satu sama lain dan diuji bersama-sama. Ini adalah untuk memastikan setiap sekumpulan komponen-komponen fungsian boleh dikaitkan antara satu sama lain tanpa sebarang masalah. Proses pengujian dilakukan berulang kali sehinggalah tidak timbul sebarang ralat dalam operasi-operasi gabungan tersebut. Misalnya dalam segmen kawalan, semua segmen kod yang telah diuji dengan baik menjamin kepantasan dan kebolehpercayaan system serta memudahkan ujian ke atas integrasi system.

Contohnya pengujian semula elemen-elemen dalam satu menu misalnya *Element Menu* dalam sistem perisian tersebut. Setiap elemen diuji supaya berfungsi seperti yang dinyatakan.

7.5 Pengujian Integrasi

Pengujian ini melibatkan pengujian terhadap satu sistem lengkap di mana komponen-komponen individu digabungkan dan dikombinasikan. Sistem dilihat sebagai satu hierarki komponen di mana setiap komponen dimiliki oleh satu lapisan rekabentuk, ia dapat memberi satu gambaran yang sebenar apabila berlakunya kegagalan system. Ralat yang mungkin wujud pada pengujian ini adalah seperti aturcara ikon yang tidak berfungsi dengan betul dan berkesan dan sebagainya. Empat pendekatan yang boleh digunakan dalam pengujian Integrasi ialah

1. Pengujian Atas-Bawah-Pengujian dijalankan pada paras yang paling atas dan seterusnya semua komponen yang dipanggil komponen yang telah diuji tadi akan diuji sebagai unit yang bergabung.
2. Pengujian Bawah-Atas-Terbalik daripada proses Atas-Bawah, di mana pengujian pada paras yang lebih rendah kemudian diikuti pada paras yang lebih tinggi.
3. Pengujian Big-Bang-Semua komponen diuji secara berasingan dan kemudian digabungkan sebagai satu sistem akhir.
4. Pengujian Sandwich-Pengujian yang menggabungkan strategi atas bawah dengan bawah atas. Sistem dilihat mempunyai TIGA lapisan:
 - Lapisan sasaran : TENGAH
 - Lapisan di atas sasaran
 - Lapisan di bawah sasaran

Bagi pengujian jenis ini, pendekatan Atas-Bawah diaplikasikan pada lapisan atas manakala pendekatan Bawah-Atas digunakan pada lapisan bawah. Pengujian ini menggunakan gabungan kelebihan pendekata Atas-Bawah dan Bawah-Atas.

Pengujian Integrasi pakej bergrafik bagi geometri ini telah menggunakan pendekatan pengujian Integrasi Sandwich kerana teknik ini menggabungkan kelebihan kaedah pengujian atas-bawah dan bawah-atas. Pengujian jenis ini membolehkan pengujian dilakukan pada peringkat lebih awal dan komponen diuji secara bersendirian atau bergabung.

7.6 Pengujian Sistem

Pengujian sistem dilakukan untuk menentukan sistem yang telah dibangunkan memenuhi spesifikasi keperluan. Terdapat 2 jenis pengujian sistem iaitu pengujian fungsi dan pengujian pencapaian. Pengujian fungsi berdasarkan keperluan fungsi sistem dan lebih difokuskan kepada fungsi-fungsi sesuatu aspek. Pengujian pencapaian pula lebih menumpu kepada keperluan yang bukan fungsian terhadap sesuatu aspek. Ia mengesahkan sesuatu fungsi dalam sistem berjalan lancar dan sistem mencapai objek dan beroperasi dengan baik. Contohnya pengujian sistem bagi pakej ini adalah dengan menginputkan data ujian seperti melukis bentuk-bentuk secara suka hati yang dibekalkan oleh sistem, menekan butang-butang dan sebagainya untuk menguji sama ada input yang dimasukkan menghasilkan output yang diinginkan. Pengujian sistem dilakukan oleh pembangun sistem sendiri dengan menguji dan memastikan yang :

- Pelaksanaan sistem tidak tergantung
- Maklumat yang dipersembahkan ringkas, padat dan menarik
- Setiap skrin dapat memanggil semula menu berkenaan , bagi mendapatkan maklumat yang berikutnya.

7.7 Pengujian Penerimaan

Pengujian penerimaan terakhir sebelum sistem diterima untuk penggunaan. Pengujian ini melibatkan pandangan daripada bakal pengguna dan orang lain selain pembangun sistem. Penguji sistem perlu mengisi borang soal-selidik penggunaan sistem seperti yang dilampirkan dalam Appendix A. Daripada borang ini, pembangun dapat mengetahui sama ada pengguna berpuas hati dengan sistem yang dibangunkan. Pengujian terhadap pandangan pengguna adalah penting untuk memastikan bahawa pengguna:

1. Dapat menggunakan sistem dengan mudah.
2. Dapat memahami arahan-arahan yang diberikan.
3. Tidak keliru dengan ikon-ikon.
4. Dapat memahami maklumat yang dipaparkan dalam skrin.

7.8 Perancangan Pengujian

Perancangan pengujian haruslah dilakukan dengan lengkap dan teliti untuk mengesan kegagalan sistem di samping memastikan sistem yang dibangunkan dapat berfungsi secara berkesan. Terdapat beberapa langkah penting dalam menentukan pengujian ke atas sistem seperti yang dinyatakan:

➤ Penetapan objektif pengujian

Setiap objek pengujian ke atas modul atau submodul ditetapkan berdasarkan fungsi-fungsi yang dilaksanakan oleh unit-unit yang hendak diuji .

➤ Merekabentuk Pengujian

Sebelum pengujian, struktur data dalam seluruh unit program tersebut haruslah diketahui untuk merekabentuk jenis ujian yang akan dijalankan. Ia dapat menguji pengesahan struktur data di samping menguji ralat yang timbul.

➤ Pelaksanaan Ujian ke atas sistem

Setelah mengenalpasti struktur data bagi ujian yang dilakukan, data-data ini dapat digunakan dalam unit system yang hendak diuji.

➤ Penilaian keputusan pengujian system

Selepas pelaksanaan ujian system dijalankan dengan lengkap, keputusan-keputusan ujian akhir yang diperoleh dianalisis untuk melihat sama ada keputusan itu mencapai objektif pengujian yang ditetapkan atau tidak. Jika objektif ujian telah dicapai, kod aturcara perlu diubah semua sehingga objektif tercapai.

7.9 Penyelenggaraan

Penyelenggaraan merupakan fasa terakhir dalam pembinaan system. Fasa ini dilakukan bagi memastikan system sentiasa dapat beroperasi dalam keadaan yang baik. Fasa ini melibatkan pembetulan dibuat ke atas ralat yang dikesan dan didapati bermasalah. Terdapat beberapa ciri penyelenggaraan iaitu:

➤ Penyelenggaraan Pembetulan

Penyelenggaraan jenis ini dibuat apabila terdapat perubahan yang dilakukan pada kod. Penyelenggaraan ini dibuat untuk membetulkan kesilapan yang tidak dapat dikesan semasa pembangunan.

➤ Penyelenggaraan Penyesuaian

Penyelenggaraan ini dibuat mengikut keperluan semasa. Ia dibuat untuk menyesuaikan keperluan yang berubah. Ia juga dijalankan sama ada mengubahsuai kod yang sedia ada ataupun menambah kod ke dalam program.

➤ Penyelenggaraan Penyempurnaan

Penyelenggaraan penyempurnaan dilakukan untuk mendapat program yang lebih baik dan lebih berkualiti. Secara amnya, proses penyelenggaraan dilakukan setelah sistem telah dicuba dan dilarikan oleh pengguna pada persekitaran yang sebenar. Selepas proses penyelenggaraan siap, dokumentasi disediakan untuk menyenangkan kanak-kanak yang bakal menggunakan system tanpa sebarang masalah. Manual pengguna yang lengkap disediakan meliputi setiap fungsian-fungsian yang ada pada system supaya pengguna dapat mengguna system dengan mudah.

7.10 Ringakasan Bab 7

Dalam pengujian, kesilapan dan ralat dikenalpastikan serta keberkesanan sesuatu aturcara menjalankan fungsi yang ditetapkan dilakukan. Tujuannya adalah untuk memastikan sistem telah mencapai objektif yang telah dikenalpasti dan memenuhi keperluan pengguna. Objektif menjalankan pengujian adalah untuk mengenalpasti ralat dan menjalankan ujian regresi.

Pengujian yang telah dijalankan ialah pengujian unit, pengujian modul, pengujian integrasi, pengujian system dan pengujian penerimaan. Pengujian ini dijalankan oleh pembangun sendiri dan hanya pengujian penerimaan dijalankan dengan meminta bantuan bakal pengguna dan orang lain secara pembangun. Selepas pengujian, penyelenggaraan dijalankan untuk memperbaiki kesilapan yang dapat dikesan.

BAB 8 PENILAIAN SISTEM DAN CADANGAN

8.1 Pengenalan

Bab ini akan membincangkan kelebihan serta kekangan-kekangan yang terdapat pada system yang dibangunkan. Cadangan-cadangan yang didapati hasil daripada pengujian sistem dan cadangan daripada pelbagai pihak diselitkan untuk mempertingkatkan lagi kualiti pakej ini pada masa hadapan. Proses penilaian telah dilakukan oleh pembangun sendiri untuk memaklumkan kepada pengguna dengan lebih jelas mengenai kekuatan, kelemahan, had serta peningkatan masa hadapan pakej ini.

8.2 Kekuatan Sistem

- Antaramuka pengguna bergrafik yang menarik

Antaramuka Pakej Bergrafik bagi Geometri ini dibangunkan berdasarkan prinsip rekabentuk antaramuka bergrafik memandangkan pengguna sasaran adalah kanak-kanak sekolah rendah. Ini adalah kerana ia mudah dipelajari dan digunakan walaupun bagi pengguna yang kurang biasa dengan sistem komputer dengan memanipulasikan prinsip-prinsip di bawah:

➤ Ciri-ciri GUI (antaramuka pengguna bergrafik)	➤ Penjelasan
➤ Tetingkap (Windows)	➤ Beberapa tetingkap yang membenarkan pemaparan maklumat yang berbeza tetapi secara serentak dalam skrin pengguna.
➤ Ikon	➤ Ikon membezakan jenis maklumat

	<p>jenis-jenis maklumat yang berbeza.</p> <p>Contohnya dalam sistem-sistem tertentu, ikon fail mewakili fail manakala ikon process mewakili proses.</p>
➤ Menu	➤ Komen dipilih dari menu berbanding dengan komen yang diinput secara menaip.
➤ Penunjuk (pointing)	➤ Satu peranti penunjuk contohnya tetikus digunakan untuk memilih komen yang diingini.
➤ Grafik	➤ Elemen grafik dapat digabungkan dengan teks bagi pemaparan yang sama.

Jadual 8.1 Ciri-ciri antaramuka yang baik

➤ Gaya Interaksi yang baik

- ✓ Manipulasi terus: Pengguna berinteraksi terus dengan objek di tettingkap. Contohnya pengguna hendak melukis bulatan, ikon bulatan yang terdapat pada tool bar diklik. Seterusnya pengguna boleh melukis di tettingkap melukis dengan hanya 'drag' tetikus tersebut.

- ✓ Pemilihan menu: Pengguna memilih arahan daripada senarai menu. Contohnya, jika pengguna ingin membuka fail baru, 'New' di dalam menu fail dipilih dan sistem akan terima komen untuk membuka fail yang baru.

➤ Cara pembelajaran yang berbeza

Pakej ini sebagai perisian melukis selain dapat membantu kanak-kanak sekolah rendah dalam mengenali beberapa bentuk asas, juga dapat membolehkan mereka memahami sistem koordinat dan pengiraan sudut sesuatu bulatan. Di samping itu, beberapa contoh aplikasi bentuk geometry digunakan dalam kehidupan seharian untuk mewakili maklumat juga dibekalkan seperti carta pie dan graf garis. Cara pembelajaran ini diharapkan dapat menarik perhatian pelajar mencuba sesuatu yang baru dan seterusnya memupuk minat mereka ke atas matapelajaran matematik.

➤ Mengautomasikan fungsi penting pakej pembelajaran

Pakej bergrafik ini dapat mengautomasikan pakej pembelajaran berbanding dengan pembelajaran secara manual melalui buku. Contohnya seperti kemudahan pengiraan luas dan perimeter bentuk-bentuk, pemaparan kedudukan bentuk-bentuk tertentu dalam sistem koordinat dan contoh-contoh bergrafik yang dibekalkan..

➤ Rekabentuk skrin yang berwarna-warni

Memandangkan pengguna sasaran utama adalah kanak-kanak, rekabentuk skrin dihasilkan dengan pelbagai warna supaya dapat menarik minat para pelajar.

8.3 Kekangan Sistem

- Sasaran pengguna yang terhad

Perisian ini hanya sesuai bagi pengguna sekolah rendah yang berumur antara 10-12 tahun sahaja. Pengetahuan yang terdedah kepada pengguna hanya sesuai bagi peringkat umur tersebut.

- Tidak dapat dicapai secara berangkaian

Sistem ini merupakan aplikasi 'stand-alone' dan tidak dapat diakses secara internet. Maka pengguna yang dapat menggunakan perisian ini adalah terhad.

8.4 Peningkatan Pada Masa Depan

Dipertingkatkan supaya dapat dicapai secara internet

Di samping dijadikan sebagai aplikasi 'stand-alone', diharapkan sistem ini juga dapat diperkembangkan sebagai 'Applet' supaya dapat digunakan melalui pengaksesan internet.

- Pengembangan skop

Contohnya, peningkatan sistem kepada kebolehan melukis graf dan bentuk tiga dimensi supaya tidak terhad kepada ciri-ciri yang tertentu sahaja.

- Penggunaan pangkalan data

Pakej ini harus dipertingkatkan kepada penggunaan pangkalan data supaya dapat menyimpan soalan-soalan latihan mengikut kategori dan jawapan. Ini adalah supaya

pakej ini dapat dijadikan satu pakej yang lengkap dalam proses pengajaran dan pembelajaran.

8.5 Masalah-masalah yang dihadapi

➤ Penguasai bahasa pengaturcaraan yang kurang

JAVA merupakan bahasa pengaturcaraan berorientasikan objek. Namun kurang pendedahan kepada bahasa ini menyebabkan masa yang agak lama diperlukan untuk pembelajaran suatu bahasa pengaturcaraan tersebut. Ini sebanyak sedikit akan mengganggu kelancaran pembangunan sistem.

Penyelesaian:

Bentuk penyelesaian yang disebabkan oleh kurang pendedahan kepada bahasa pengaturcaraan baru adalah dengan meminjam buku rujukan mengenai bahasa pengaturcaraan Java dari perpustakaan sebagai proses pembelajaran, melayari internet untuk mendapat bantuan dan mendapatkan nasihat daripada pensyarah .

➤ Langkah 'SET PATH' diperlukan setelah instalasi (installation) JDK1.3.1

Masalah yang dihadapi adalah setelah instalasi JDK1.3.1 , percubaan mengkompilkan program di MS Command Prompt dengan 'javac Sketcher.java' tidak dapat dijalankan kerana tidak disedari bahawa langkah 'SET PATH' diperlukan.

Penyelesaian:

Langkah 'SET PATH=C:\jdk1.3.1\bin' perlu dilakukan sama ada di MS Command Prompt ataupun mengikut langkah di bawah:

1. 'Right click' tetikus dengan menunjuk ke ikon My Computer di desktop.
2. Dari menu yang 'popup' itu , 'Properties' dipilih.

3. Setelah itu, satu kotak dialog akan muncul dan butang 'Advanced' dipilih.
4. Selepas itu, masuk ke 'Environment Variables'. Di sana, butang 'New' diklik.
5. Masukkan 'SET PATH=C:\jdk1.3.1\bin' ke dalam kotak teks yang disediakan.
6. Kemudian klik butang 'OK'.

➤ Masalah lain yang tidak diingini

Terdapat beberapa masalah yang tidak diingini telah dihadapi semasa pembangunan sistem ini seperti PC sendiri diserangi oleh virus.

Penyelesaian

Membersihkan (clean) fail-fail yang mempunyai virus dengan menggunakan perisian antivirus.

8.6 Cadangan-cadangan

- Tugas yang diberikan kepada pelajar pada tahun akhir yang perlu menyiapkan latihan ilmiah iaitu subjek lain perlu dikurangkan. Ini membolehkan mereka menumpukkan perhatian terhadap projek tahap akhir ini.
- Tugas dan tutorial yang banyak bagi setiap subjek menyebabkan masa diperuntukkan terhadap sistem adalah sedikit dan pelajar terpaksa mengecilkan skop sistem pada saat akhir bagi membolehkan projek akhir ini disiapkan.
- Masa yang diperuntukkan untuk menggunakan kemudahan bilik dokumen juga perlu diperpanjangkan dan selaras dengan waktu perpustakaan utama dan tidak dihadkan hanya untuk waktu-waktu tertentu dan pada hari-hari tertentu sahaja. Ini menyukarkan

pelajar untuk mencari rujukan di sana kerana banyak masa dihabiskan untuk menghadiri kuliah serta diperuntukkan untuk membuat tugas.

- Terdapat beberapa projek tahun akhir yang berpotensi untuk dikomersilkan. Oleh itu, pihak fakulti perlu mengambil langkah dan memberi sokongan akan kerja-kerja ini dapat diketengahkan dan ini memberi peluang kepada pelajar untuk menonjolkan diri mereka di kalangan masyarakat.

8.7 Ringkasan Bab 8

Berdasarkan penilaian yang telah dibuat, pakej perisian bergrafik yang dibangunkan itu mempunyai kelebihan dan kekurangannya.

Diharapkan kelebihan yang ada dikekalkan dan kekurangan yang ada harus dipertingkatkan. Melalui pelaksanaan penilaian, pembangun sistem dapat menyedari kekurangan sistem yang dibangunkan. Daripada pengalaman ini, diharapkan dapat membangunkan sistem yang lebih baik pada masa depan.

8.8 Kesimpulan

1. Sistem yang telah dibangun ini dapat mengautomasikan fungsi-fungsi sebuah pakej pembelajaran manual dan dapat mencapai matlamat yang ditetapkan.
2. Beberapa perkara telah dipelajari sepanjang pembangunan sistem ini seperti pengurusan masa yang baik dan mempelajari penggunaan perisian baru iaitu JCreator LE 2.0 dan bahasa pengaturcaraan JAVA untuk membangunkan sistem. Walau bagaimanapun, pelbagai masalah dihadapi seperti kurang penguasaan yang cukup terhadap bahasa pengaturcaraan JAVA, kesuntukan masa dan dibebani pula dengan tugas-tugas subjek lain selain menyelesaikan projek ilmiah tahap akhir ini.
3. Namun demikian, pelaksanaan projek ini telah dapat melatih untuk berdepan dengan masalah-masalah yang sukar dan kurang yakin dapat menyiapkan projek ini serta apa yang telah dilakukan.
4. Latihan ini juga telah memberi peluang untuk mempraktikkan apa yang telah diikuti sepanjang program Sarjana Muda Sains Komputer. Contohnya, penyelenggaraan dan pembangunan sistem yang dipelajari dalam subjek Kejuruteraan Perisian, Analisis dan Rekabentuk Sistem Maklumat, Rekabentuk Antaramuka Bergrafik (GUI), teknik-teknik pengaturcaraan dan sebagainya.
5. Setelah melalui pengalaman untuk menyiapkan projek ilmiah ini, pengetahuan memprogram aturcara dan kemahiran pengaturcaraan dapat dipertingkatkan di samping dapat mempelajari penggunaan bahasa pengaturcaraan JAVA yang semakin popular ini.
6. Pelaksanaan latihan ilmiah ini juga dapat memberi pengalaman untuk membangunkan sebuah sistem secara individu dan persendirian. Ini

LAM meningkatkan keyakinan diri sebelum menempuh alam pekerjaan yang lebih mencabar.

1. Adakah anda mempunyai komputer?

Ya
Tidak

2. Adakah anda menggunakan komputer?

Ya
Tidak

3. Adakah anda tahu apa itu internet melalui komputer?

Ya
Tidak

4. Bagaimanakah anda menilai komputer yang teras?

5. Apakah tahap penguasaan anda menggunakan komputer?

1	2	3	4	5
Tidak	Kurang	Cukup	Terampil	Terampil
Tahap				
tinggi				

6. Apakah minat anda terhadap dunia perisian (termasuk dalam MS Word)?

1	2	3	4	5
---	---	---	---	---

LAMPIRAN A

Borang soal kaji selidik

1. Adakah anda tahu apa itu komputer?

<input type="checkbox"/>	Ya
<input type="checkbox"/>	Tidak

2. Adakah anda mempunyai computer?

<input type="checkbox"/>	Ya
<input type="checkbox"/>	Tidak

3. Adakah anda tahu apa itu belajaran melalui computer?

<input type="checkbox"/>	Ya
<input type="checkbox"/>	Tidak

Bagi soalan 4 & 5 , sila bulatkan pilihan yang sesuai:

4. Apakah tahap pengetahuan anda menggunakan computer?

1 2 3 4 5

Tidak Memuaskan Sangat tahu

tahu

langsung

5. Apakah minat anda terhadap mata pelajaran Geometri dalam Matematik?

1 2 3 4 5

Tidak Memuaskan Sangat tahu

suka

langsung

Bagi soalan yang seterusnya, sila tandakan petak yang sesuai:

6. Pernahkah anda menggunakan pakej pembelajaran berbantuan komputer

Ya

Tidak

7. Adakah anda ingin memilih cara pembelajaran yang berteknologi seperti penggunaan computer sebagai alat pembelajaran?

Ya

Tidak

8. Apakah pendapat anda tentang keberkesanan pembelajaran antara penggunaan papan hitam dan computer?

Papan hitam lebih berkesan

Komputer lebih berkesan

Kedua-duanya sama sahaja

Tiada

komen

9. Pada pendapat anda, adakah pakej ini akan memberikan banyak manfaat di dalam pembelajaran apabila ia siap dibangunkan kelak?

Ya

Tidak

10. Sebagai kesimpulan, adakah anda menyokong penghasilan pakej pembelajaran ini?

Ya

Tidak

LAMPIRAN B

Manual Pengguna

Kandungan

Pengenalan

Bab 1: Permulaan Penggunaan Sistem

1.1 Keperluan-keperluan sistem 120

1.2 Memulakan perisian 120

Bab 2: Penjelasan Penggunaan Perisian

2.1 Menu 122

2.1.1 Menu Fail 122

2.1.2 Menu Elemen 123

2.1.3 Menu Options 123

2.1.4 Menu Tool 124

2.1.5 Menu Help 125

2.2 Bebutang di Toolbar 125

2.3 Menu Popup 128

2.4 Menu Popup untuk aksi-aksi dalam bentuk yang dipilih 131

Bab 3 Keterangan bagi aplikasi-aplikasi tertentu 133

3.1 Melukis teks 133

3.2 Menggunakan kalkulator 133

3.3 Melukis Polygon 134

3.4 Bar status 135

3.5 Memusing, mentranslasi dan penskalaan segitiga 136

3.6 Melukis 'Arc' dengan sudut-sudut yang telah dispesifikasikan 137

3.7 Mengubahsuaikan bentuk elips dengan menginputkan nilai

tinggi dan lebar

138

3.8 Pengiraan perimeter

139

3.9 Pengiraan luas

140

3.10 Memilih font dari kotak font

141

3.11 Memilih ketebalan garis

141

University of Malaya

Manual Pengguna

Pengenalan

Pakej bergrafik bagi geometri merupakan satu perisian yang dihasilkan untuk membantu para pelajar sekolah rendah dalam mengenali beberapa bentuk geometri yang asas. Di samping itu, penggunaan pakej perisian ini juga dapat meningkatkan kreativiti pelajar-pelajar sekolah rendah kerana pengguna dapat menghasilkan bentuk yang diinginkan dengan menginput koordinat titik-titik tertentu dan mengisi bentuk dengan warna yang diinginkan.

Selain itu, beberapa informasi yang berguna kepada para pengguna seperti carta pie, graf garis, maklumat-maklumat asas bagi bentuk dua dimensi dan maklumat-maklumat bagi bentuk tiga dimensi juga disampaikan kepada pelajar.

Sungguhpun, perisian ini bukan satu pakej yang lengkap dalam proses pengajaran dan pembelajaran, tetapi memadai menyampaikan pengetahuan yang penting kepada pengguna.

Bab 1 Permulaan Pengguna Pakej Bergrafik bagi Geometri :Sketcher

1.1 Keperluan-keperluan sistem

Berikut adalah keperluan-keperluan minimum yang diperlukan untuk menjalankan perisian

Pakej Bergrafik bagi Geometri: Sketcher.

i) Keperluan perkakasan

- Komputer peribadi dengan
- Pemprosesan Pentium
- Saiz ingatan 64 SDRAM
- Cakera keras (hard disk) 10GB
- Tetikus
- Monitor 15"
- Kad grafik

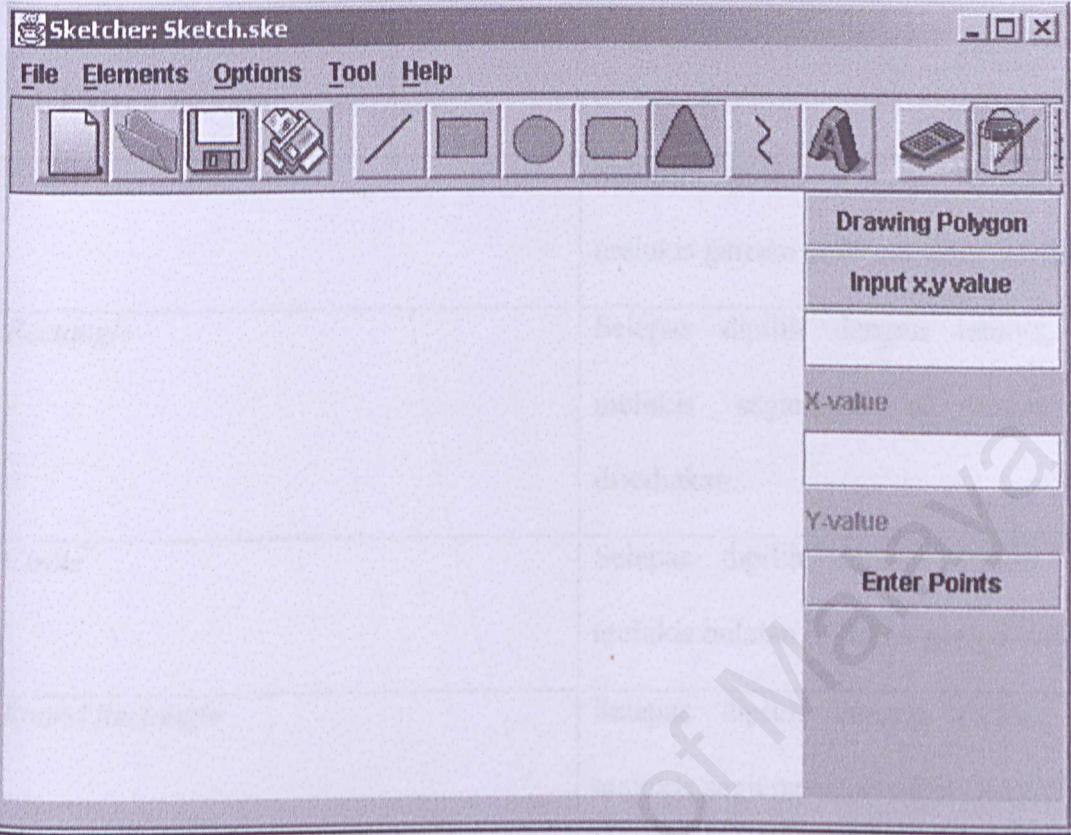
ii) Keperluan perisian

Microsoft Windows 2000-Professional Edition

1.2 Memulakan perisian

Perisian dimulakan dengan dengan mengklik fail yang dilabelkan 'run', maka perisian akan dapat digunakan secara terus.

Bab 2 Penjelasan penggunaan Pakej Bergrafik bagi Geometri: Sketcher



Rajah 2.1 Skrin utama Sketcher

2.1 Menu

2.1.1 Menu Fail

<i>New</i> (Ctrl + N)	Membuka fail yang baru
<i>Open</i> (Ctrl + s)	Membuka fail dari folder yang menyimpan fail tersebut
<i>Save</i> (Ctrl + N)	Menyimpan fail
<i>Save As</i>	Menyimpan fail dengan bentuk fail dengan pilihan ‘file extension’ dan tempat yang diinginkan
<i>Print</i> (Ctrl + N)	Mencetak sesuatu fail

<i>Exit</i> (Ctrl + N)	Keluar dari sistem
------------------------	--------------------

2.1.2 Menu Elemen

<i>Line</i>	Selepas dipilih dengan tetikus, boleh melukis garisan di frame yang disediakan.
<i>Rectangle</i>	Selepas dipilih dengan tetikus, boleh melukis segiempat di frame yang disediakan.
<i>Circle</i>	Selepas dipilih dengan tetikus, boleh melukis bulatan di frame yang disediakan.
<i>Round Rectangle</i>	Selepas dipilih dengan tetikus, boleh melukis segiempat bersudut bulat di frame yang disediakan.
<i>Curve</i>	Selepas dipilih dengan tetikus, boleh melukis lengkung di frame yang disediakan.
<i>Text</i>	Selepas dipilih dengan tetikus, boleh menulis teks di frame yang disediakan. [Rujuk Bahagian 3.1]

2.1.3 Menu Options

<i>Choose Font...</i>	Memilih bentuk font yang ingin digunakan. [Rujuk Bahagian 3.10]
-----------------------	---

<i>Line Width</i>	Memilih ketebalan garis [Rujuk Bahagian 3.11].
<i>Fill Options (Solid)</i>	Setelah dipilih, bentuk yang dilukis adalah telah diisi warna.
<i>Fill Options (outline)</i>	Setelah dipilih, bentuk yang dilukis hanya terdiri daripada garis luaran tanpa diisi warna.
<i>Notes (2D Shapes)</i>	Menunjukkan skrin mempunyai keterangan ringkas mengenai bentuk 2 dimensi.
<i>Notes (3D Shapes)</i>	Menunjukkan skrin mempunyai keterangan ringkas mengenai bentuk 3 dimensi.
<i>EllipseTest</i>	Skrin menguji ketinggian (height) dan kelebaran (width) elips. [Rujuk Bahagian 3.7]
<i>Perimeter Calculation</i>	Skrin menunjukkan cara pengiraan perimeter bulatan dan segiempat. [Rujuk Bahagian 3.8]
<i>Area Calculation</i>	Skrin menunjukkan cara pengiraan luas bulatan dan segiempat. [Rujuk Bahagian 3.9]





2.1.4 Menu Tool












<i>Calculator</i>	Setelah dipilih, kalkulator akan 'pop-up' keluar dan dapat digunakan untuk membantu dalam pengiraan.
<i>Draw Arc</i>	Setelah dipilih, satu tettingkap akan 'pop-up' keluar dan dapat melukis 'arc' mengikut kesesuaian yang dihendaki. [Rujuk Bahagian 3.6]







2.1.5 Menu Help

<i>About</i>	Memaparkan informasi mengenai sistem
--------------	--------------------------------------

2.2 Bebutang di ToolBar

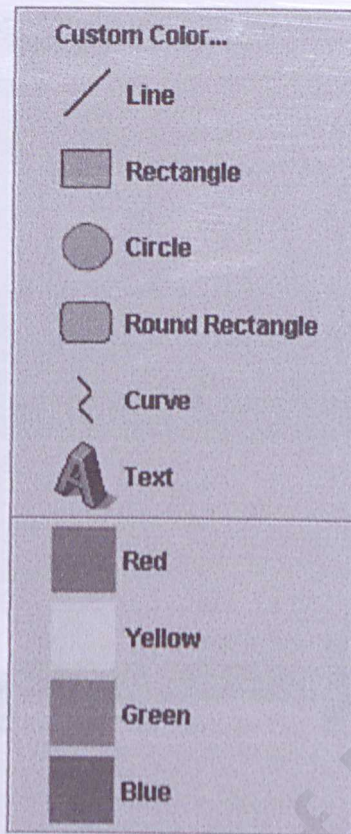
Ikon	Penjelasan
	Memulakan tingkap yang baru.
	Membukan folder yang menyimpan fail atau membuka semula fail yang telah disimpan.
	Menyimpan fail-fail yang telah dicipta. Contohnya ingin menyimpan lukisan yang dihasilkan.
	Mencetak data-data yang ingin

	dipersembahkan.
	Butang ini ditekan apabila ingin melukis garisan.
	Butang ini ditekan apabila ingin melukis segiempat.
	Butang ini ditekan apabila ingin melukis bulatan.
	Butang ini ditekan apabila ingin melukis segiempat bersudut bulat.
	Butang ini ditekan apabila ingin memaparkan tetingkap segitiga. [Rujuk Bahagian 3.5]
	Butang ini ditekan apabila ingin melukis lengkung.
	Butang ini ditekan apabila ingin mengiput teks. [Rujuk Bahagian 3.1]
	Kalkulator akan 'pop-up' keluar apabila butang ini ditekan dan pengiraan boleh dilakukan. [Rujuk Bahagian 3.2]
	Butang ini ditekan apabila ingin melukis bentuk yang telah diisikan dengan warna.
	Memaparkan graph garis lengkung.

	Memaparkan carta pie.
	Memaparkan nota-nota pelajaran.
	Butang bagi warna merah.
	Butang bagi warna kuning.
	Butang bagi warna hijau.
	Butang bagi warna biru.

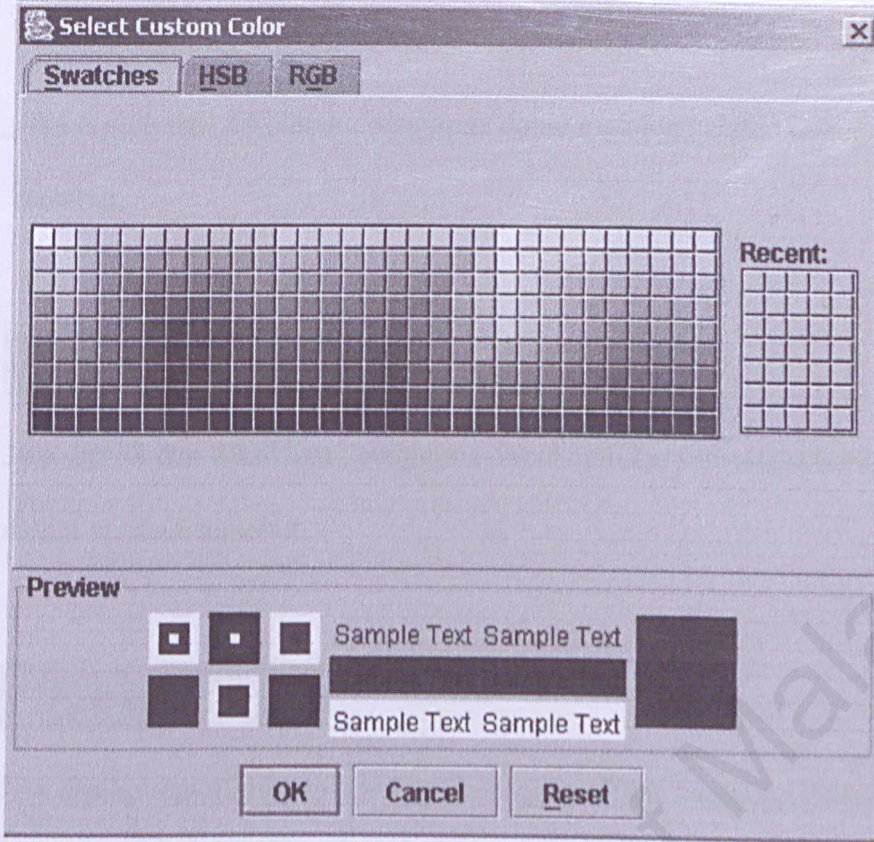
2.3 Menu Popup

Selain menggunakan menu yang disediakan untuk memilih element yang hendak dilukis atau digunakan, pengguna juga dapat '*right klik*' tetikus untuk menggunakan menu popup yang disediakan tetapi dengan syarat kawasan dipilih untuk melukis tidak diisi dengan sebarang lukisan. Menu popup yang disediakan seperti yang ditunjukkan di bawah:

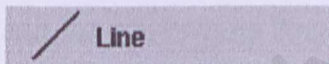


Custom Color...

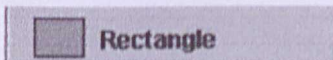
Apabila 'Custom Color...' diklikkan, satu palette untuk memilih warna akan dipopup keluar seperti berikut:



1. Pilih warna yang dikehendaki.
2. Setelah memilih warna yang dipilih, klik Ok.



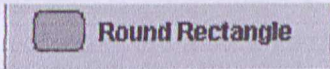
Jika dipilih dan diklikkan , pengguna dapat melukis garis lurus dalam aplikasi tersebut.



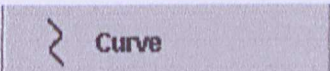
Jika dipilih dan diklikkan , pengguna dapat melukis segiempat dalam aplikasi tersebut.



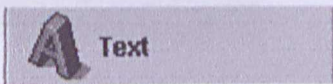
___ Jika dipilih dan diklikkan , pengguna dapat melukis bulatan dalam aplikasi tersebut.



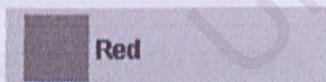
Jika dipilih dan diklikkan , pengguna dapat melukis segiempat bersudut lengkung dalam aplikasi tersebut.



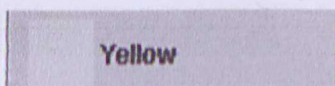
Jika dipilih dan diklikkan , pengguna dapat melukis lengkung dalam aplikasi tersebut.



Jika dipilih dan diklikkan , pengguna dapat melukis teks dalam aplikasi tersebut (operasi seperti yang diterangkan di bahagian 3.1).



Jika dipilih dan diklikkan , pengguna dapat melukis dalam warna merah.



Jika dipilih dan diklikkan , pengguna dapat melukis dalam warna kuning.

A rectangular button with a dark green square on the left and the word "Green" in black text on the right.

Green


Jika dipilih dan diklikkan , pengguna dapat melukis dalam warna hijau.

A rectangular button with a dark blue square on the left and the word "Blue" in black text on the right.

Blue

Jika dipilih dan diklikkan , pengguna dapat melukis dalam warna biru.

2.4 Menu Popup untuk aksi-aksi bagi bentuk yang dipilih:

A vertical list of four menu items: "Move", "Delete", "Rotate", and "Send-to-back". Each item is on a separate line, and the entire list is enclosed in a light gray rectangular box.

Move
Delete
Rotate
Send-to-back

A rectangular button with the word "Move" in black text.

Move

Jika dipilih dan diklikkan , pengguna dapat menggerakkan bentuk yang dilukis dengan syarat ia telah dipilih (highlighted).

A rectangular button with the word "Delete" in black text.

Delete

Jika dipilih dan diklikkan , pengguna dapat 'delete' bentuk yang dilukis dengan syarat ia telah dipilih (highlighted).

A rectangular button with the word "Rotate" in black text.

Rotate


Jika dipilih dan diklikkan , pengguna dapat memusingkan bentuk yang dilukis dengan syarat ia telah dipilih (highlighted).

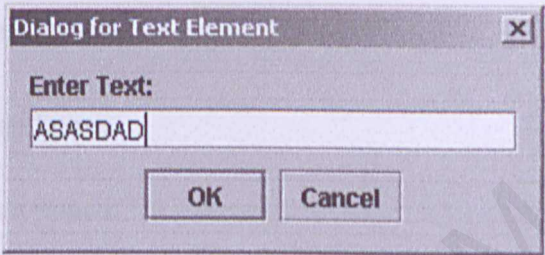
Send-to-back

Jika dipilih dan diklikkan , pengguna dapat menghantar bentuk yang dilukis ke belakang bentuk yang lain dengan syarat ia telah dipilih (highlighted).

Bab 3 Keterangan bagi aplikasi-aplikasi tertentu.


3.1 Melukis teks

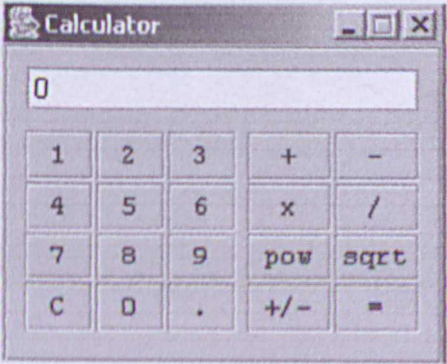
Jika pengguna ingin melukis teks , butang  akan ditekan. Selepas itu, klikkan tetikus di tempat yang ingin diletakkan teks tersebut. Satu kotak dialog seperti berikut akan keluar.



- 1.Inputkan teks ke dalam kotak teks yang disediakan
- 2.Kemudian klik butang OK.
- 3.Selepas itu , teks tersebut akan dipaparkan.

3.2 Menggunakan Kalkulator

Jika ingin menggunakan kalkulator , klikkan ikon . Satu kalkulator seperti berikut seperti berikut akan dipopup keluar.



Pengguna boleh menginput nombor ke dalam kotak teks yang disediakan untuk melakukan pengiraan.

Keterangan bagi fungsi-fungsi butang kalkulator:

<div>+</div>	operasi tambah
<div>-</div>	operasi tolak
<div>x</div>	operasi darab
<div>/</div>	operasi bahagi
<div>pow</div>	operasi kuasa
<div>sqrt</div>	operasi punca
<div>+/-</div>	Jika ingin input nombor negatif atau nombor positif
<div>=</div>	Hasil setelah operasi selesai dijalankan.

3.3 Melukiskan polygon

Drawing Polygon, Input Coordinat x, y

X-value

Y-value

Enter Points

Reset


1. Inputkan koordinat x dan koordinat y sebanyak tiga kali atau lebih jika ingin melukis polygon dan tekan *Enter Points* setiap kali setelah operasi input selesai.
2. Selepas itu bebutang *Draw Poly* akan muncul . Jika ingin paparkan polygon tersebut bebutang *Draw Poly* ditekan. Polygon akan dipaparkan window.

3.4 Bar Status

Bar Status yang memaparkan kedudukan koordinat x dan koordinat y apabila tetikus diklik atau bergerak serta paparan luas and perimeter bentuk bulatan dan segiempat yang telah dilukis.

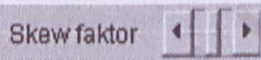
Point: [256.0,72.0] Area: 0.0 Perimeter: 0.0

3.5 Memusing, mentranslasikan dan penskalaan segitiga

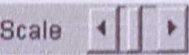
Apabila  , satu tetingkap yang memaparkan satu segitiga akan dipaparkan.



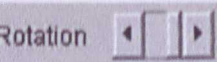
Penerangan bagi bar skrol dalam skrin:



Apabila diskrolkan, pemesongan sisi segitiga akan berubah.



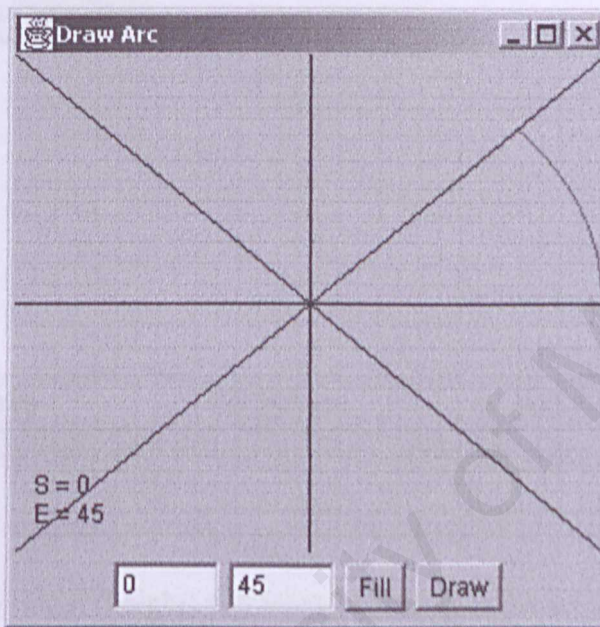
Penskalaan bagi segitiga, iaitu boleh membesarkan segitiga dan mengecilkan segitiga.



Apabila diskrolkan, seluruh segitiga boleh dipusingkan.

3.6 Melukis 'Arc' dengan sudut-sudut yang telah dispesifikasikan

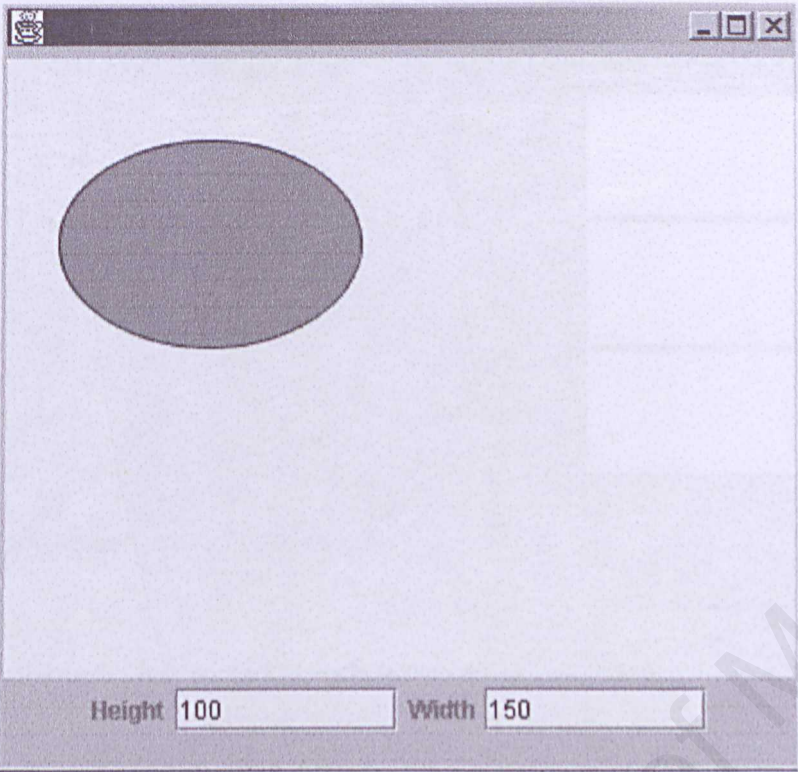
Apabila *Draw Arc* dipilih satu skrin seperti berikut akan dikeluarkan:



Jika ingin melukis 'arc',

1. Input nilai sudut mula dan sudut akhir di kotak teks yang disediakan.
2. Jika tiada perubahan nilai sudut lagi, klikkan bebutang Draw tersebut.
3. Jika ingin mengisi warna bagi bentuk yang telah dilukis, tekan bebutang Fill.

3.7 Mengubahsuai bentuk ellipse dengan menginputkan nilai tinggi dan lebar



- 1.Input nilai tinggi (height) dan lebar (width) di kotak teks yang disediakan.
- 2.Kemudian tekan ENTER.
- 3.Output dengan ukuran yang diinput akan dikeluarkan.

3.8 Pengiraan perimeter

Perimeter Calculation

Enter length, width or radius

Length

Width

Radius

Perimeter Circle, $2 \cdot \text{PI} \cdot \text{Radius}$, $\text{PI} = 3.14$

Perimeter Rectangle, $(2 \cdot \text{length}) + (2 \cdot \text{width})$

PERIMETER CIRCLE

PERIMETER RECTANGLE

1. Input nilai jejari (radius), tinggi (height) dan lebar (width) di kotak teks yang disediakan.
2. Kemudian klikkan butang PERIMETER CIRCLE untuk melihat keputusan perimeter bulatan..
3. Jika ingin melihat keputusan perimeter segiempat, klikkan butang PERIMETER RECTANGLE.

3.9 Pengiraan Luas

Area Calculation

Enter length, width or radius

Length

Width

Radius

Area Circle, $PI \times Radius \times Radius$, $PI=3.14$

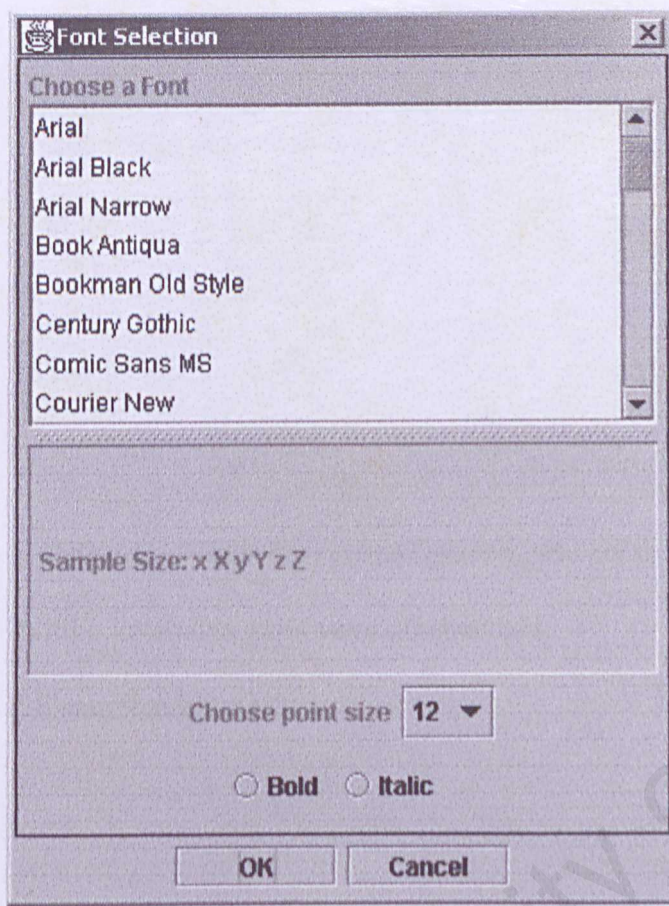
Area Rectangle, $length \times width$

AREA CIRCLE

AREA RECTANGLE

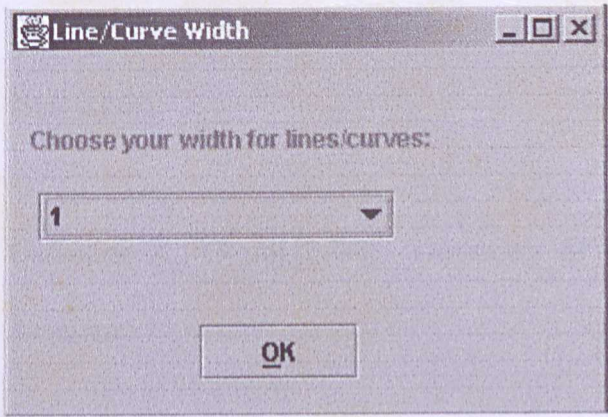
1. Input nilai jejari (radius), tinggi (height) dan lebar (width) di kotak teks yang disediakan.
2. Kemudian klikkan butang AREA CIRCLE untuk melihat keputusan luas bulatan..
3. Jika ingin melihat keputusan perimeter segiempat, klikkan butang luas RECTANGLE.

3.10 Memilih font dari kotak font



- 1..Pilih *Choose Font..* dari menu Options, satu kotak dialog font akan dipopupkan.
2. Pilih jenis font dan saiz yang dikehendaki
- 3.Pilih sama ada Bold atau Italic atau tidak bagi kedua-duanya.
- 4.Kemudian klik OK.

3.11 Memilih ketebalan garis



- 1..Pilih *Line Width* dari menu Options, satu kotak dialog font akan dipopupkan.
- 2. Pilih ketebalan garis yang dikehendaki
- 3.Kemudian klik OK.

RUJUKAN

Ali Bahrami,[1999], '*Object Oriented Systems Development*' , Irwin McGraw-Hill, New York

Catherine A., Gorini,[2000], '*Geometry At Work*', The mathematical Association of America.

Chan Yook Lean,[2002], '*TARGET UPSR Merit Matematik 2002/2003, Tahun 4, 5, 6, KBSR*' , SASBADI Sdn. , Bhd. , Petaling Jaya

Dennis Kafura, [2000], '*Object-oriented Software Design & Construction with JAVA, web enhanced*' , Prentice Hall, USA

Kendall , K.E., Kendall, J.E.,[1999], '*System Analysis And Design*' , edisi ke-4, Prentice Hall, USA

H.M.Deitel, P.J. Deitel, S.E.Santry,[2002], '*Advanced Java™ 2 Platform –How To Program—*',Prentice Hall, USA

H.M.Deitel, P.J. Deitel, [2002], '*Java™ 2 Platform –How To Program— 4th*', Prentice Hall, USA

<http://www.cs.uu.nl/~markov/kids/draw/>

<http://www.efofex.com>

<http://www.mathworks.com/products/matlab/>

<http://www.math.utah.edu/lab/ms/matlab/matlab.html>

http://www.mathsnet.net/resource/rec_software.html

Ivar Jacobson, [1999], Grady Booch, James Rumbaugh, ‘ *The Unified Software Development Process* ’, An Imprint of Addison Wesley Longman, Inc., Massachusetts

Kendall Scott, [2001] ‘UML Explained’, Addison-Wesley, New York.

Kim Topley, [1998], ‘ *CORE JavaTM Foundation Classes* ’, Prentice Hall PTR, Upper Saddle River

Mark Austin, David Chancogne, [2001], ‘ *Introduction To Engineering Programming: IN C, MATLAB and JAVA* ’, John Wiley & Sons, Inc., New York

Pfleeger, S.L., [2001],. ‘ *Software Engineering: Theory and Practice. 2nd* ’. Upper Saddle River: Prentice-Hall Inc

Roger, T. Stevens, [1999], ‘ *Graphics Programming with JavaTM, 2nd* ’, Charles River Media, INC, Rockland, Massachusetts